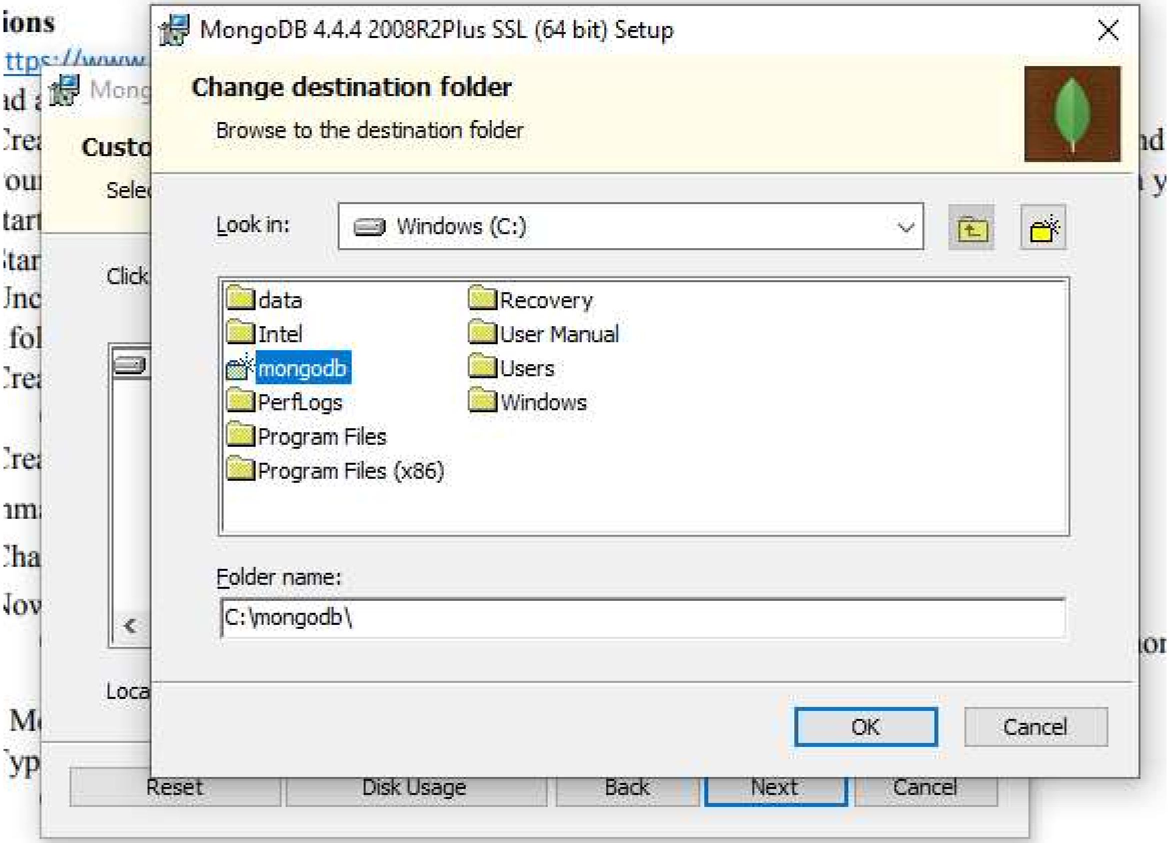
Mongodb Installation

Step 2:



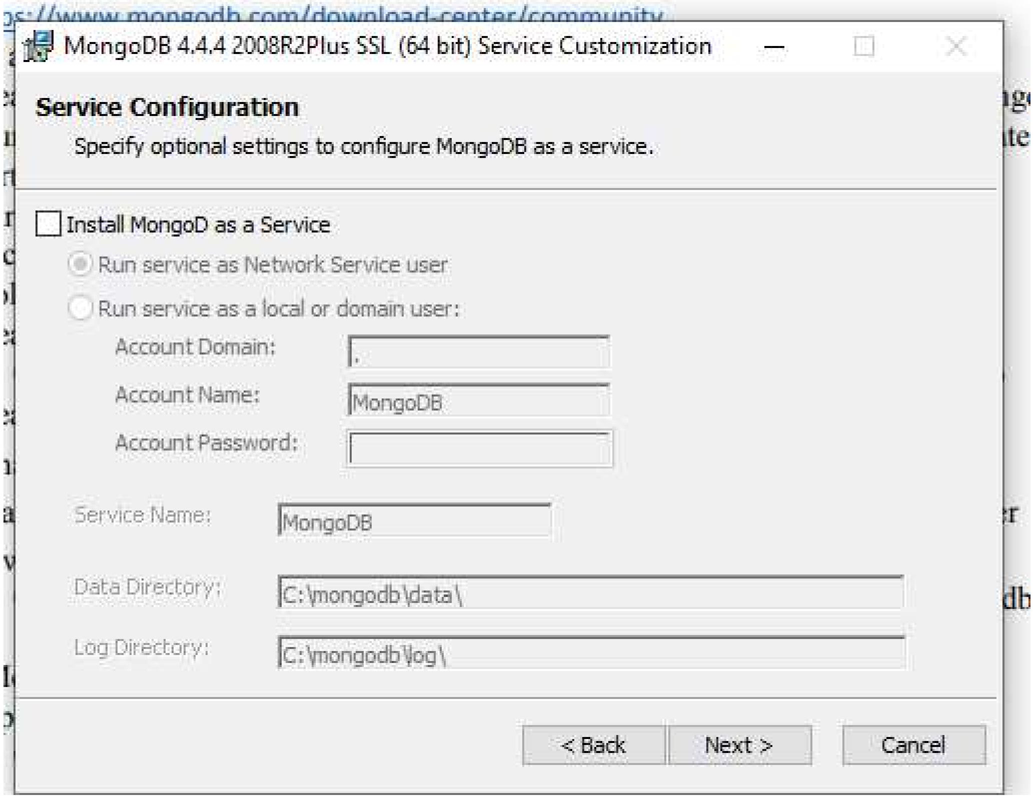
MongoDB 4.4.4 2008R2Plus SSL (64 bit) Setup

Change destination folder

Browse to the destination folder

Look in: Windows (C:)

Folder name: C:\mongodb\



ps://www.mongodb.com/download-center/community

MongoDB 4.4.4 2008R2Plus SSL (64 bit) Service Customization

Service Configuration

Specify optional settings to configure MongoDB as a service.

□ Install MongoD as a Service

● Run service as Network Service user

○ Run service as a local or domain user:

Account Domain: .

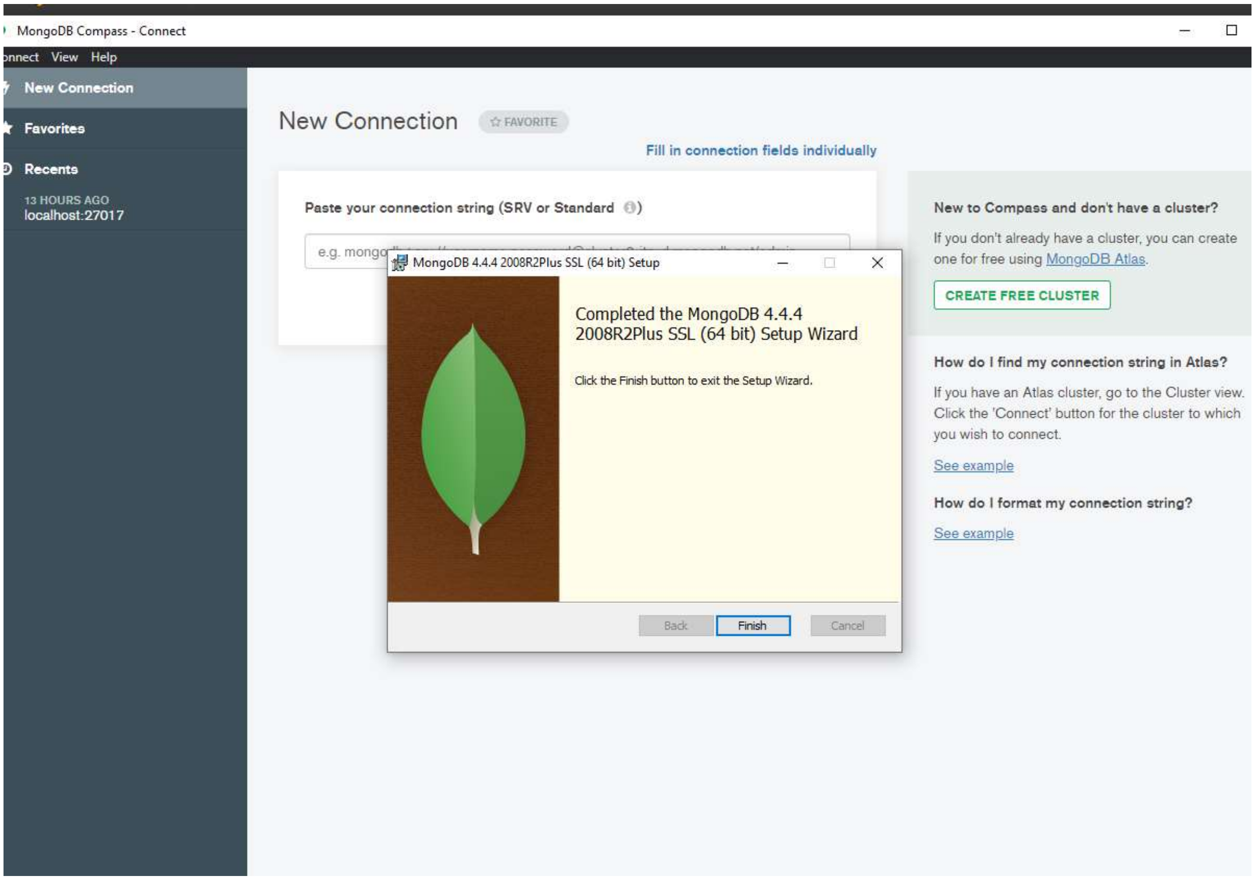
Account Name: MongoDB

Account Password:

Service Name: MongoDB

Data Directory: C:\mongodb\data\

Log Directory: C:\mongodb\log\



MongoDB Compass - Connect

13 HOURS AGO
localhost:27017

New Connection ☆ FAVORITE

Fill in connection fields individually

Paste your connection string (SRV or Standard ⓘ)

e.g. mongo

MongoDB 4.4.4 2008R2Plus SSL (64 bit) Setup

Completed the MongoDB 4.4.4 2008R2Plus SSL (64 bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using MongoDB Atlas.

How do I find my connection string in Atlas?

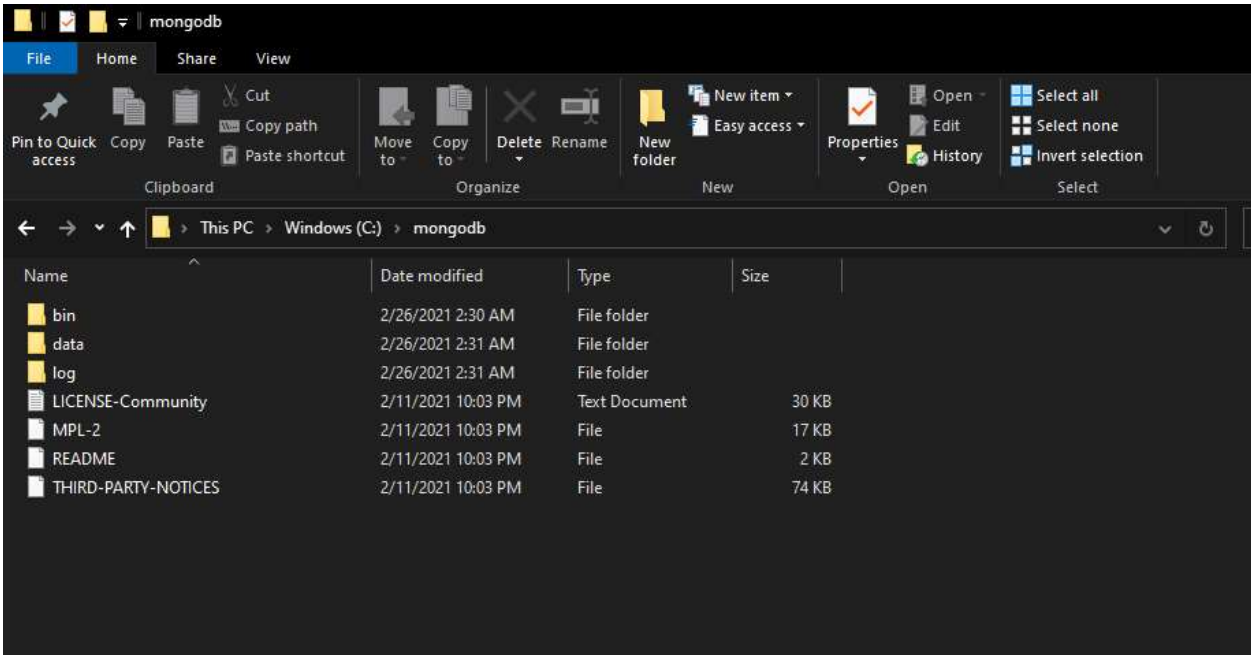
If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

See example

How do I format my connection string?

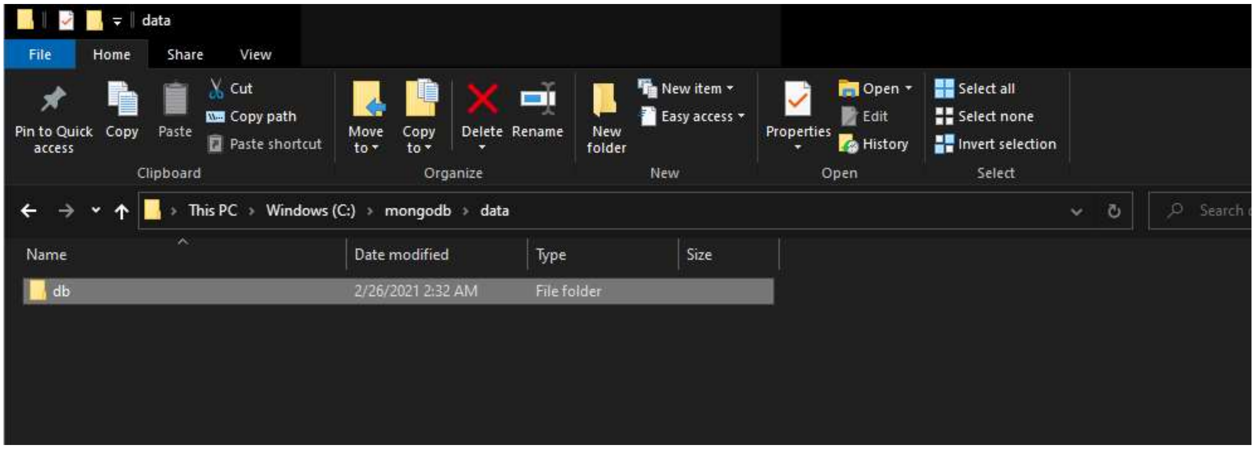
See example

Step 3:



This PC › Windows (C:) › mongodb

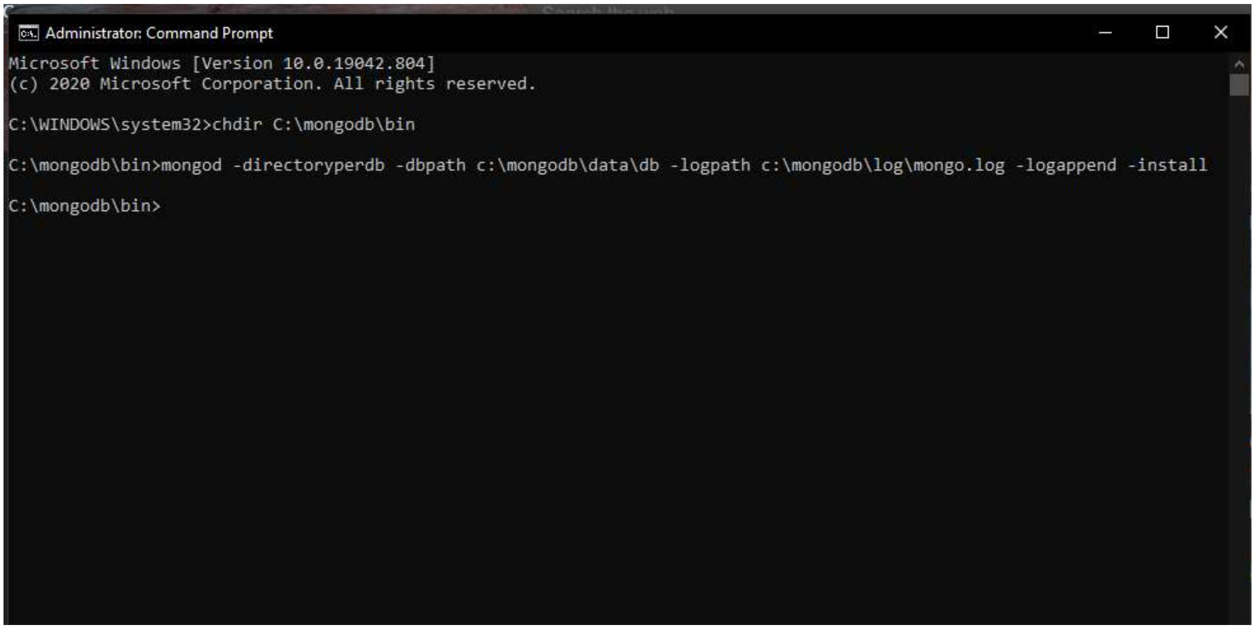
Name	Date modified	Type	Size
bin	2/26/2021 2:30 AM	File folder	
data	2/26/2021 2:31 AM	File folder	
log	2/26/2021 2:31 AM	File folder	
LICENSE-Community	2/11/2021 10:03 PM	Text Document	30 KB
MPL-2	2/11/2021 10:03 PM	File	17 KB
README	2/11/2021 10:03 PM	File	2 KB
THIRD-PARTY-NOTICES	2/11/2021 10:03 PM	File	74 KB



This PC › Windows (C:) › mongodb › data

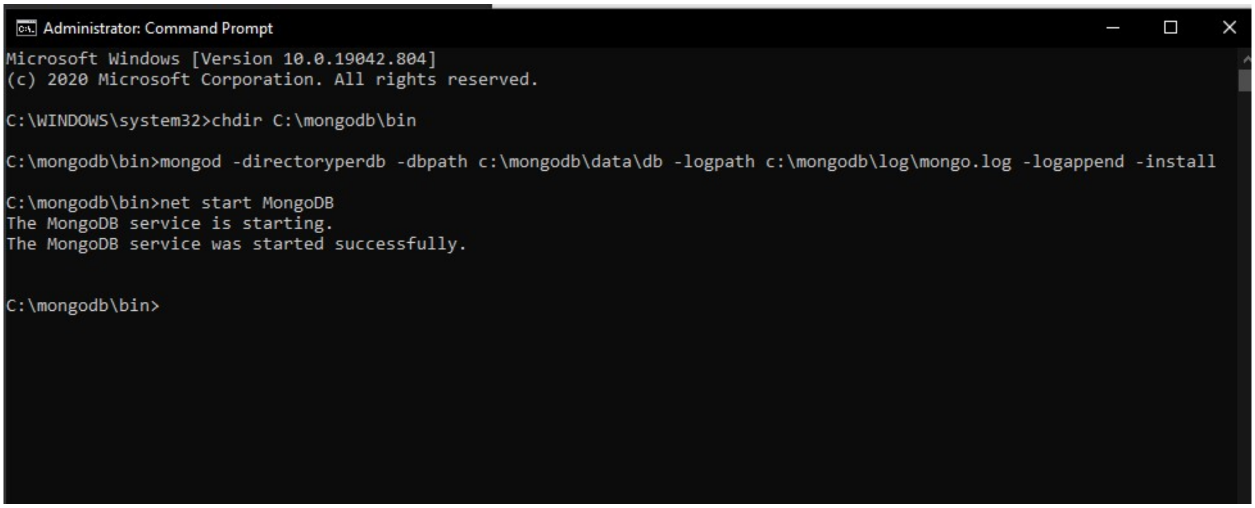
Name	Date modified	Type	Size
db	2/26/2021 2:32 AM	File folder	

Step 4:



```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>chdir C:\mongodb\bin

C:\mongodb\bin>mongod -directoryperdb -dbpath c:\mongodb\data\db -logpath c:\mongodb\log\mongo.log -logappend -install

C:\mongodb\bin>
```

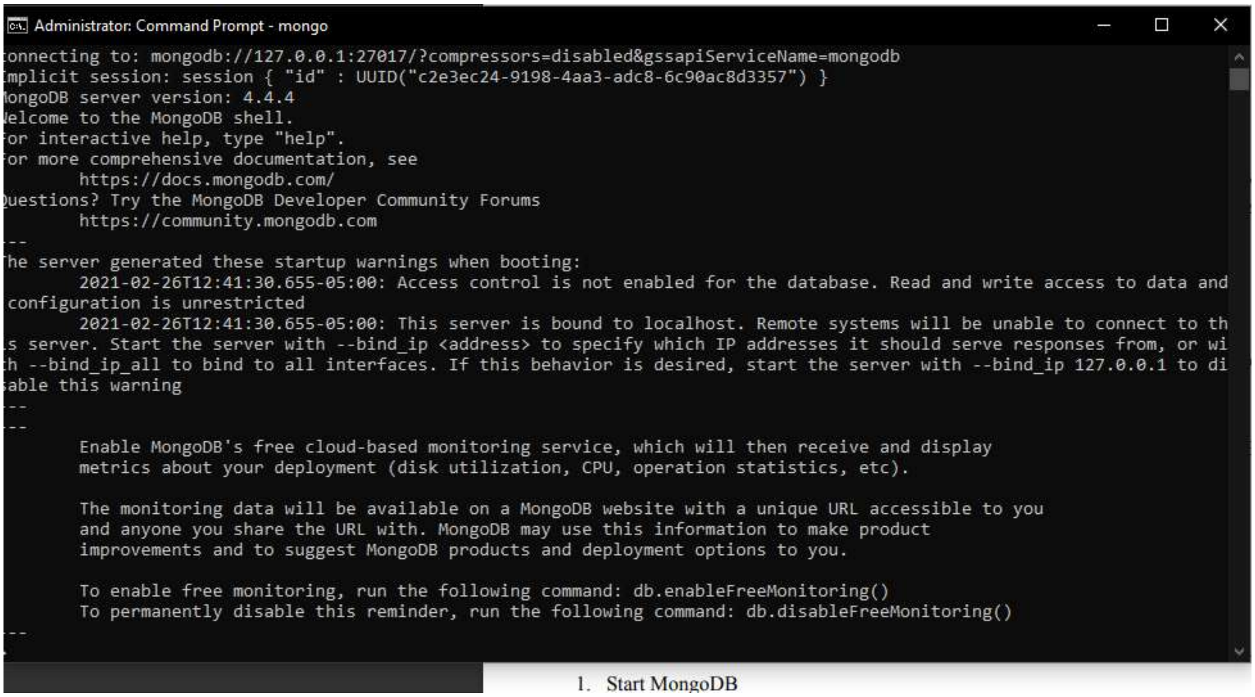
Step 5:



```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>chdir C:\mongodb\bin

C:\mongodb\bin>mongod -directoryperdb -dbpath c:\mongodb\data\db -logpath c:\mongodb\log\mongo.log -logappend -install

C:\mongodb\bin>net start MongoDB
The MongoDB service is starting.
The MongoDB service was started successfully.


C:\mongodb\bin>
```

Exercise 1:

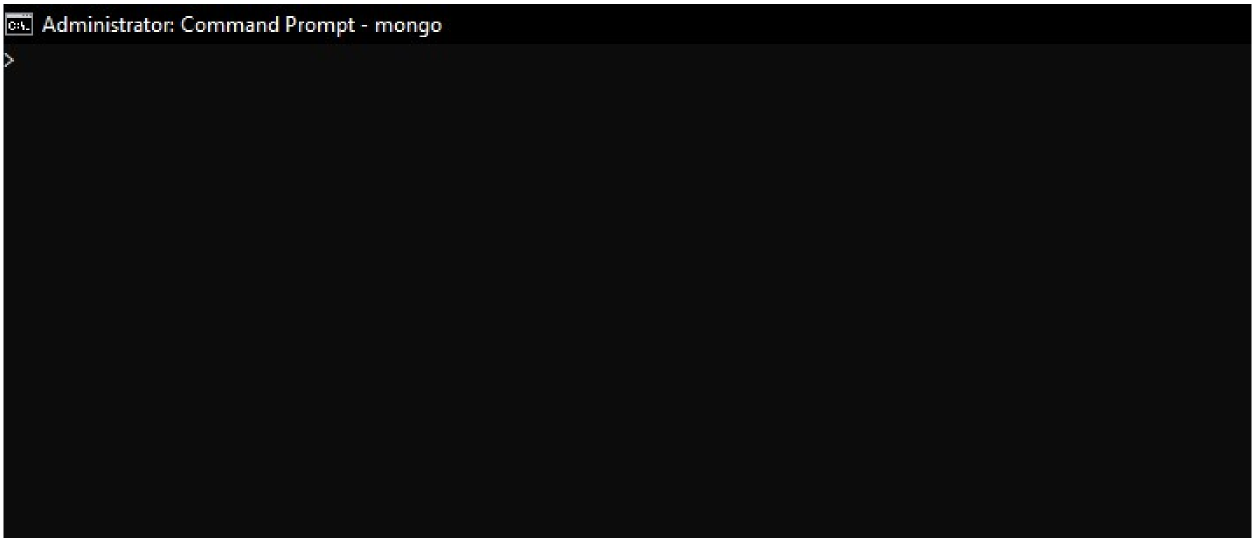
Starting mongo shell



```
onnecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
mplicit session: session { "id" : UUID("c2e3ec24-9198-4aa3-adc8-6c90ac8d3357") }
ongoDB server version: 4.4.4
elcome to the MongoDB shell.
or interactive help, type "help".
or more comprehensive documentation, see
        https://docs.mongodb.com/
uestions? Try the MongoDB Developer Community Forums
        https://community.mongodb.com
--
he server generated these startup warnings when booting:
        2021-02-26T12:41:30.655-05:00: Access control is not enabled for the database. Read and write access to data and
 configuration is unrestricted
        2021-02-26T12:41:30.655-05:00: This server is bound to localhost. Remote systems will be unable to connect to th
s server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or wi
h --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to di
able this warning
--
--
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
--
```

1. Start MongoDB

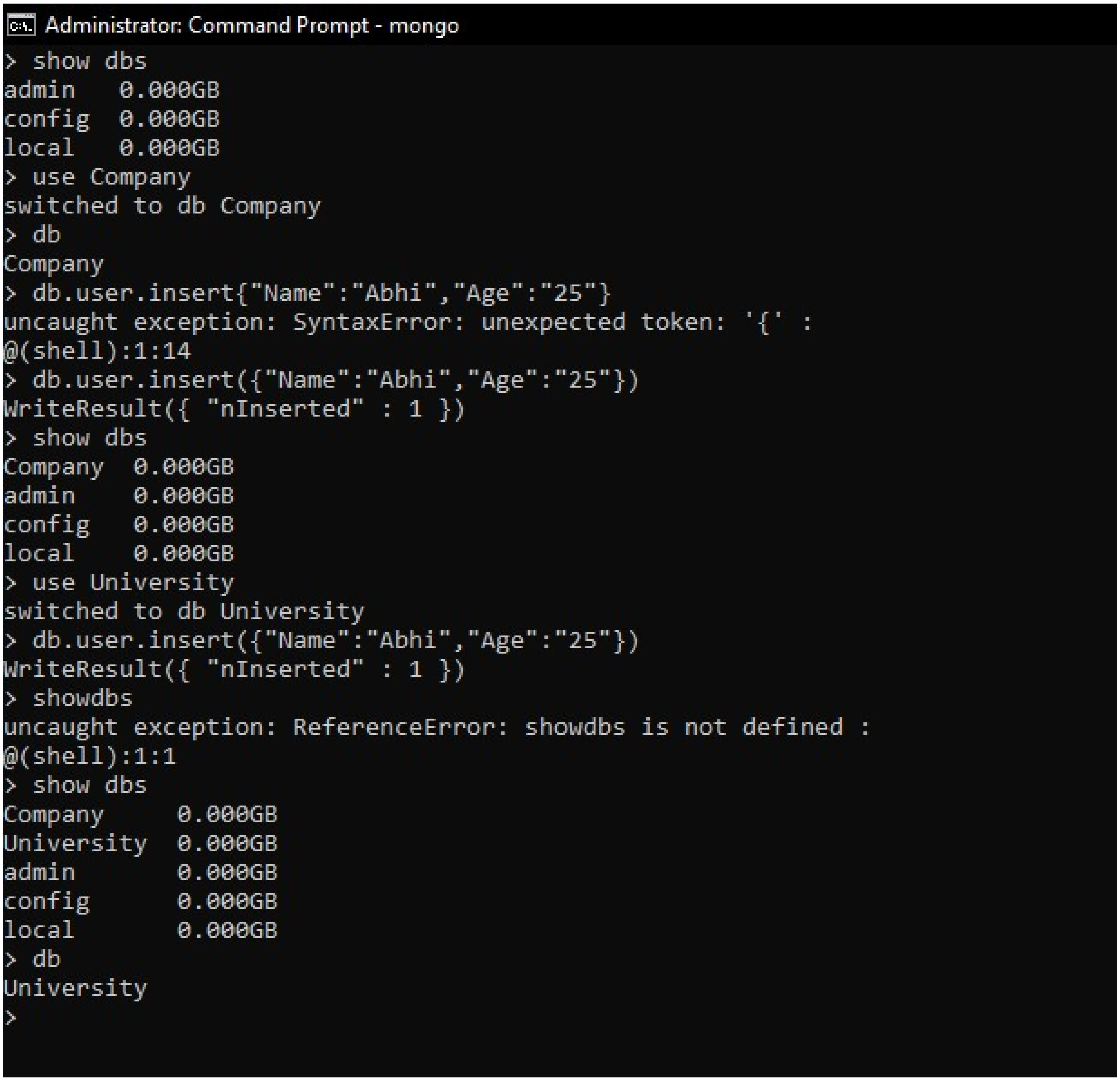
clear the screen



```
>
```

Exercise 2:

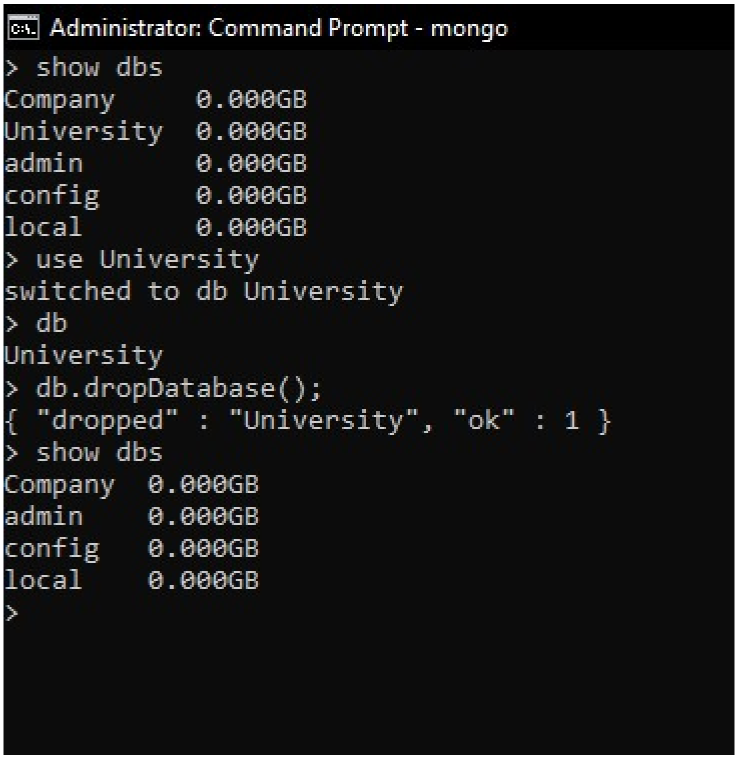
Show Databases present and creating a new Database



```
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
> use Company
switched to db Company
> db
Company
> db.user.insert{"Name":"Abhi","Age":"25"}
uncaught exception: SyntaxError: unexpected token: '{' :
@(shell):1:14
> db.user.insert({"Name":"Abhi","Age":"25"})
WriteResult({ "nInserted" : 1 })
> show dbs
Company  0.000GB
admin    0.000GB
config   0.000GB
local    0.000GB
> use University
switched to db University
> db.user.insert({"Name":"Abhi","Age":"25"})
WriteResult({ "nInserted" : 1 })
> showdbs
uncaught exception: ReferenceError: showdbs is not defined :
@(shell):1:1
> show dbs
Company     0.000GB
University  0.000GB
admin       0.000GB
config      0.000GB
local       0.000GB
> db
University
>
```

Exercise 3:

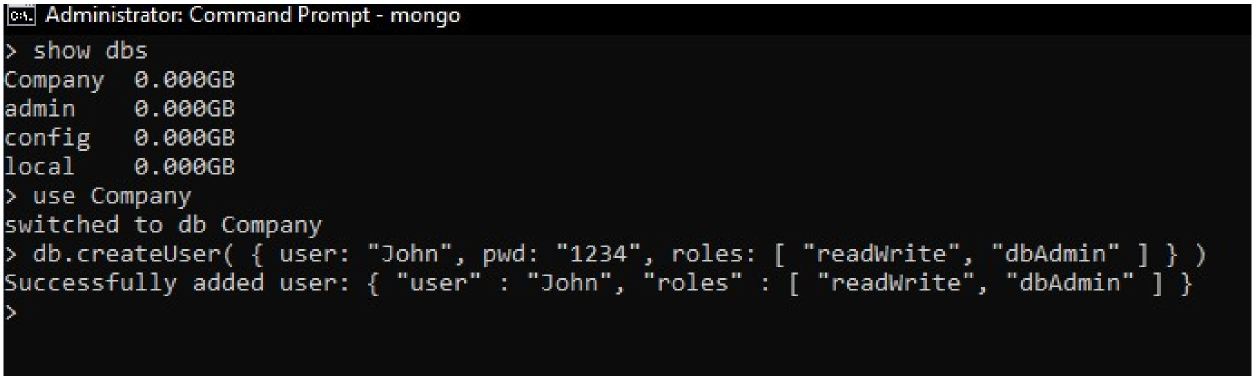
Dropping Databases



```
> show dbs
Company     0.000GB
University  0.000GB
admin       0.000GB
config      0.000GB
local       0.000GB
> use University
switched to db University
> db
University
> db.dropDatabase();
{ "dropped" : "University", "ok" : 1 }
> show dbs
Company  0.000GB
admin    0.000GB
config   0.000GB
local    0.000GB
>
```

Exercise 4:

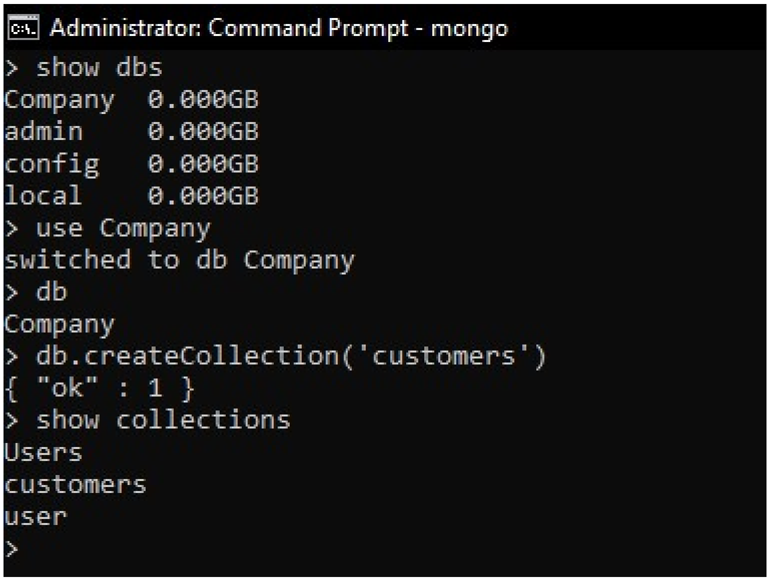
Creating user for the database “Company”



```
> show dbs
Company  0.000GB
admin    0.000GB
config   0.000GB
local    0.000GB
> use Company
switched to db Company
> db.createUser( { user: "John", pwd: "1234", roles: [ "readWrite", "dbAdmin" ] } )
Successfully added user: { "user" : "John", "roles" : [ "readWrite", "dbAdmin" ] }
>
```

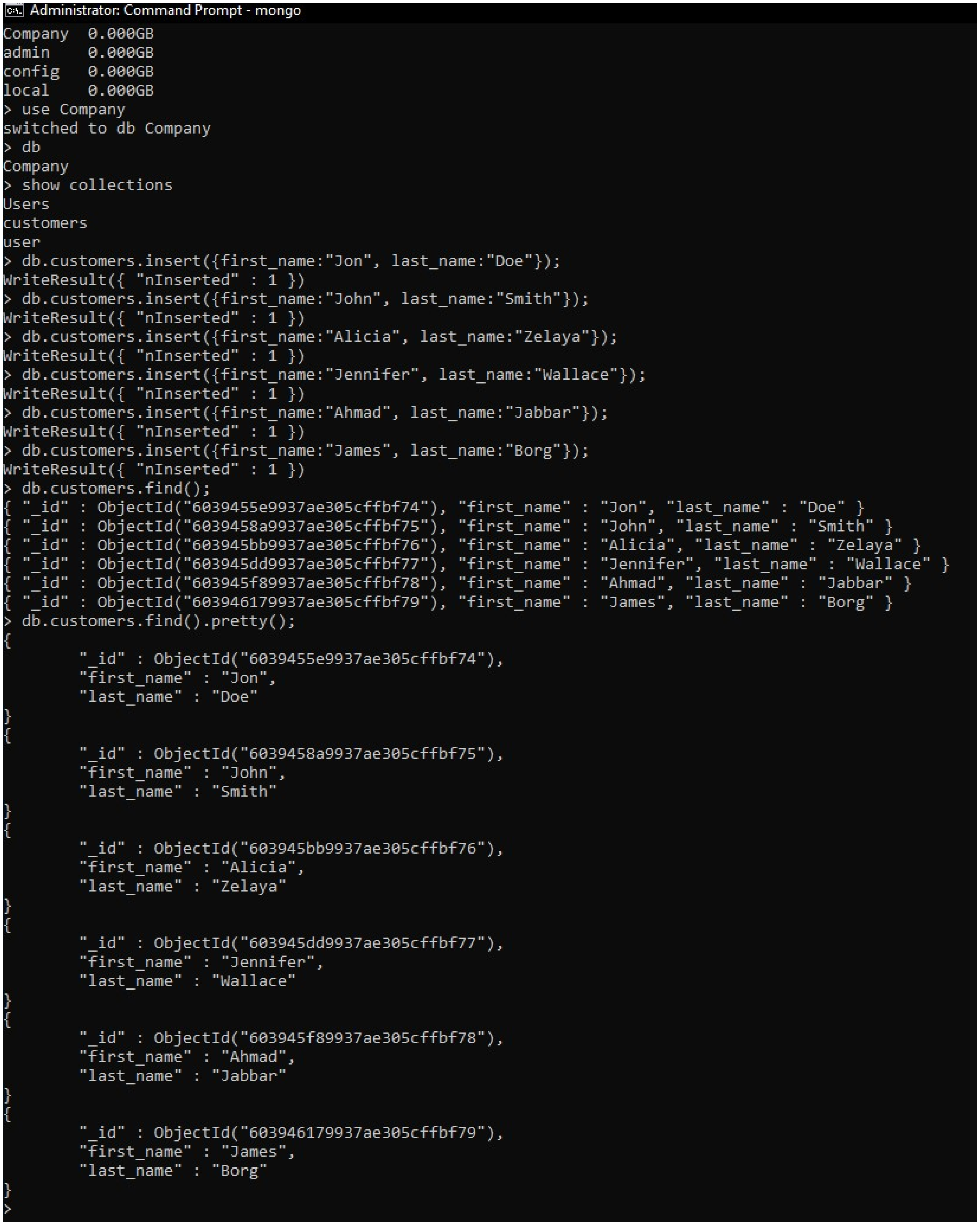
Exercise 5:

Creating and displaying Collections



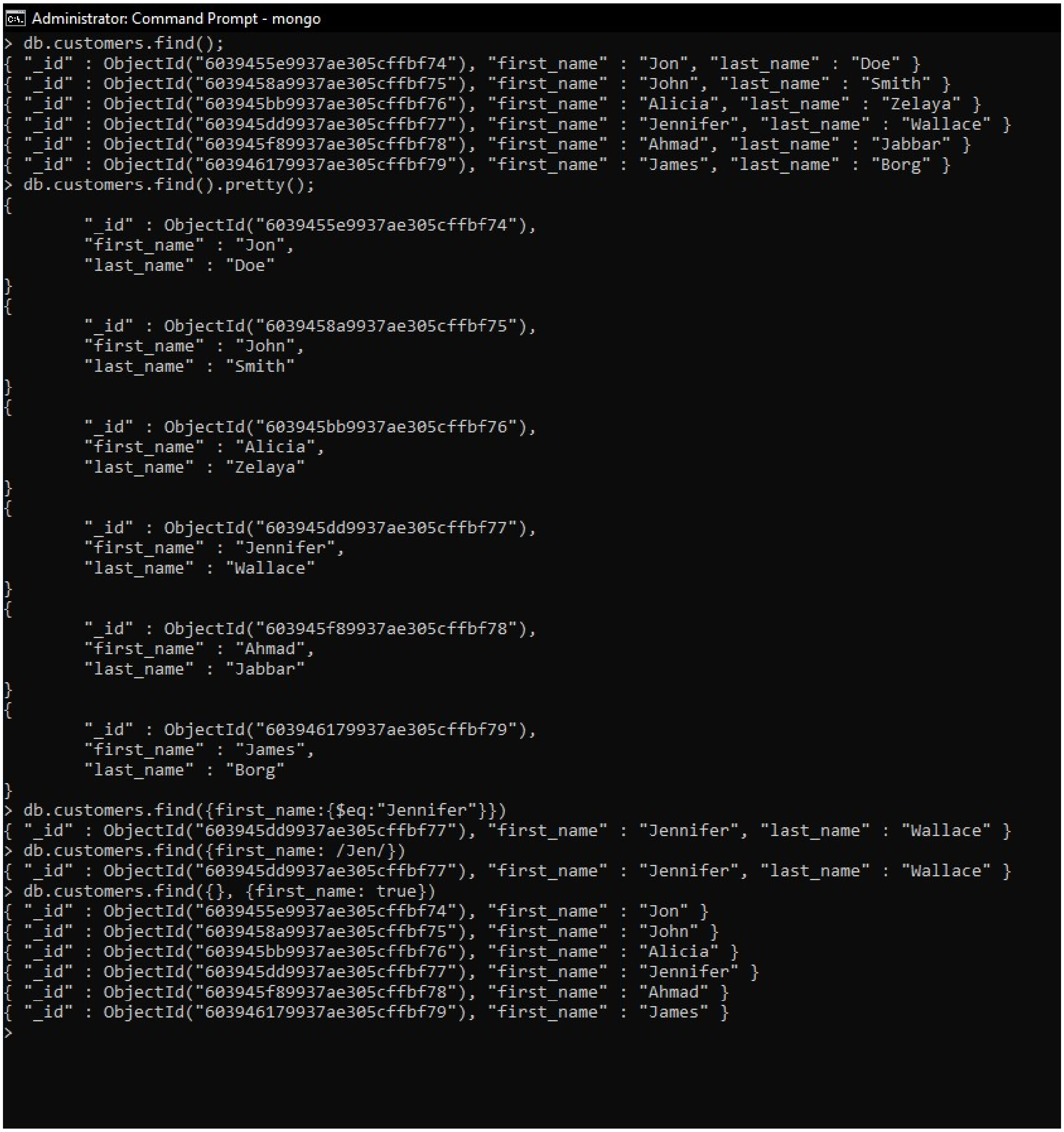
```
> show dbs
Company  0.000GB
admin    0.000GB
config   0.000GB
local    0.000GB
> use Company
switched to db Company
> db
Company
> db.createCollection('customers')
{ "ok" : 1 }
> show collections
Users
customers
user
>
```

Exercise 6:



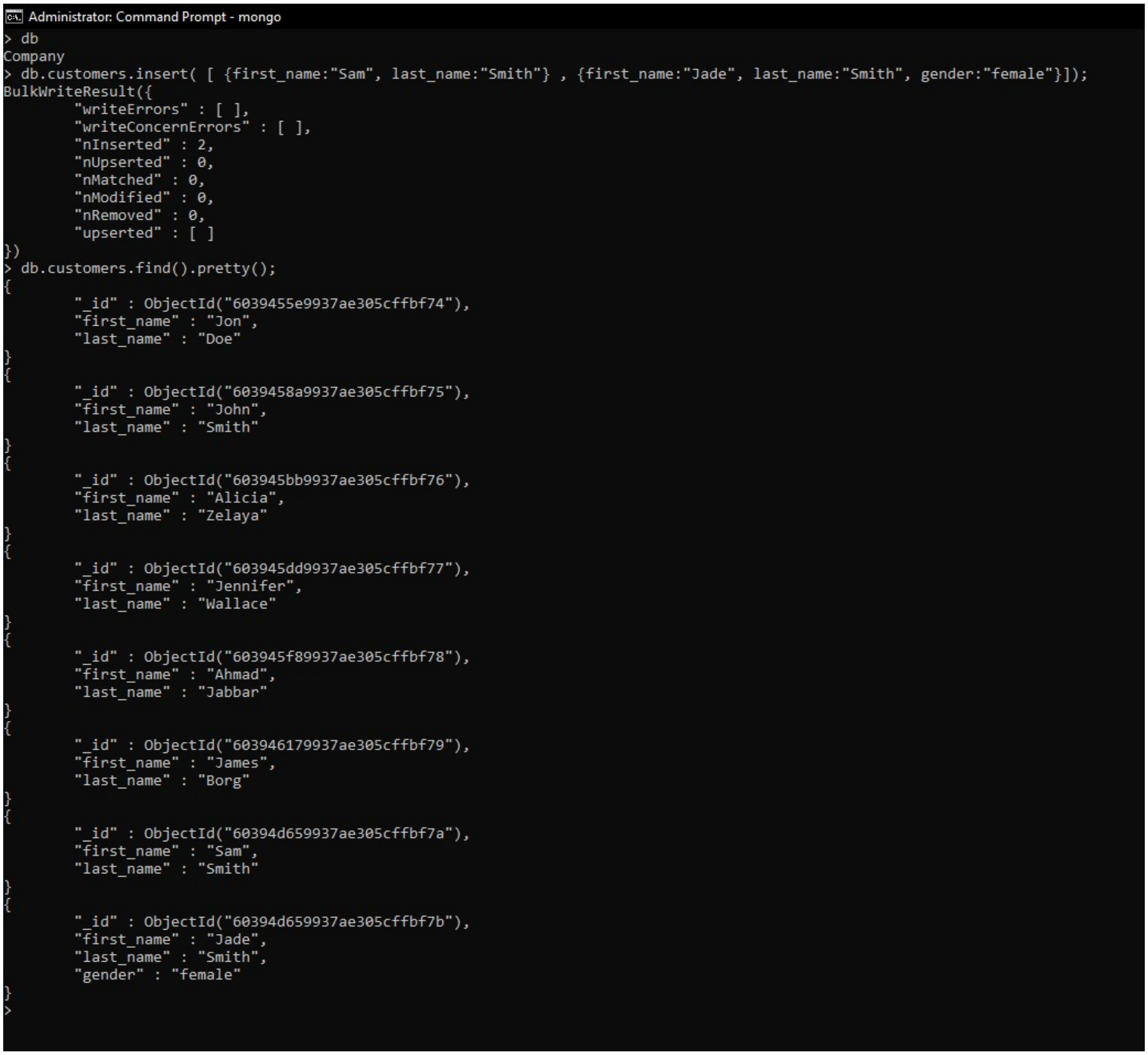
```
Company  0.000GB
admin    0.000GB
config   0.000GB
local    0.000GB
> use Company
switched to db Company
> db
Company
> show collections
Users
customers
user
> db.customers.insert({first_name:"Jon", last_name:"Doe"});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({first_name:"John", last_name:"Smith"});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({first_name:"Alicia", last_name:"Zelaya"});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({first_name:"Jennifer", last_name:"Wallace"});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({first_name:"Ahmad", last_name:"Jabbar"});
WriteResult({ "nInserted" : 1 })
> db.customers.insert({first_name:"James", last_name:"Borg"});
WriteResult({ "nInserted" : 1 })
> db.customers.find();
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe" }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith" }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya" }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace" }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar" }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg" }
> db.customers.find().pretty();
{
        "_id" : ObjectId("6039455e9937ae305cffbf74"),
        "first_name" : "Jon",
        "last_name" : "Doe"
}
{
        "_id" : ObjectId("6039458a9937ae305cffbf75"),
        "first_name" : "John",
        "last_name" : "Smith"
}
{
        "_id" : ObjectId("603945bb9937ae305cffbf76"),
        "first_name" : "Alicia",
        "last_name" : "Zelaya"
}
{
        "_id" : ObjectId("603945dd9937ae305cffbf77"),
        "first_name" : "Jennifer",
        "last_name" : "Wallace"
}
{
        "_id" : ObjectId("603945f89937ae305cffbf78"),
        "first_name" : "Ahmad",
        "last_name" : "Jabbar"
}
{
        "_id" : ObjectId("603946179937ae305cffbf79"),
        "first_name" : "James",
        "last_name" : "Borg"
}
>
```

Exercise 7:



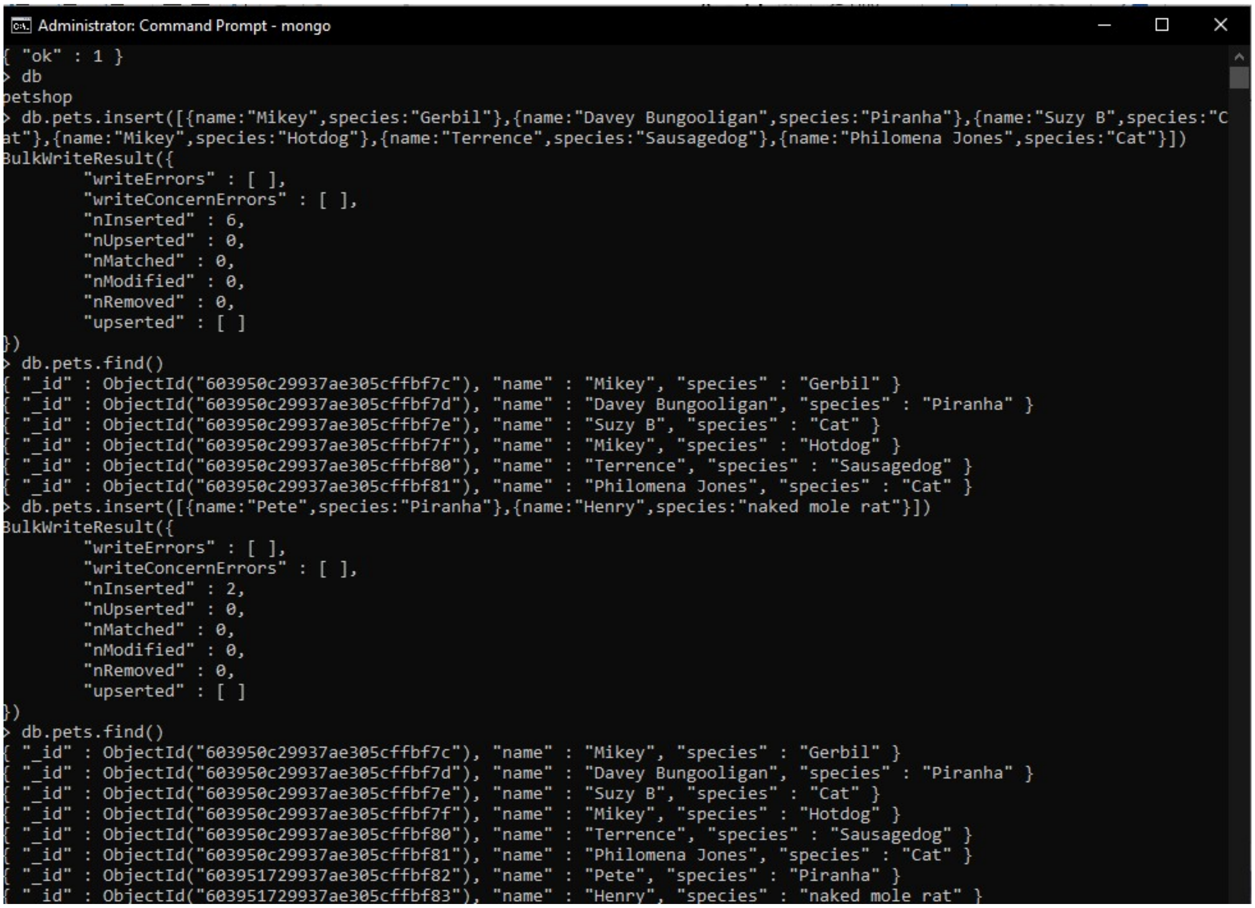
```
> db.customers.find();
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe" }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith" }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya" }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace" }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar" }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg" }
> db.customers.find().pretty();
{
        "_id" : ObjectId("6039455e9937ae305cffbf74"),
        "first_name" : "Jon",
        "last_name" : "Doe"
}
{
        "_id" : ObjectId("6039458a9937ae305cffbf75"),
        "first_name" : "John",
        "last_name" : "Smith"
}
{
        "_id" : ObjectId("603945bb9937ae305cffbf76"),
        "first_name" : "Alicia",
        "last_name" : "Zelaya"
}
{
        "_id" : ObjectId("603945dd9937ae305cffbf77"),
        "first_name" : "Jennifer",
        "last_name" : "Wallace"
}
{
        "_id" : ObjectId("603945f89937ae305cffbf78"),
        "first_name" : "Ahmad",
        "last_name" : "Jabbar"
}
{
        "_id" : ObjectId("603946179937ae305cffbf79"),
        "first_name" : "James",
        "last_name" : "Borg"
}
> db.customers.find({first_name:{$eq:"Jennifer"}})
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace" }
> db.customers.find({first_name: /Jen/})
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace" }
> db.customers.find({}, {first_name: true})
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon" }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John" }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia" }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer" }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad" }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James" }
>
```

Exercise 8:

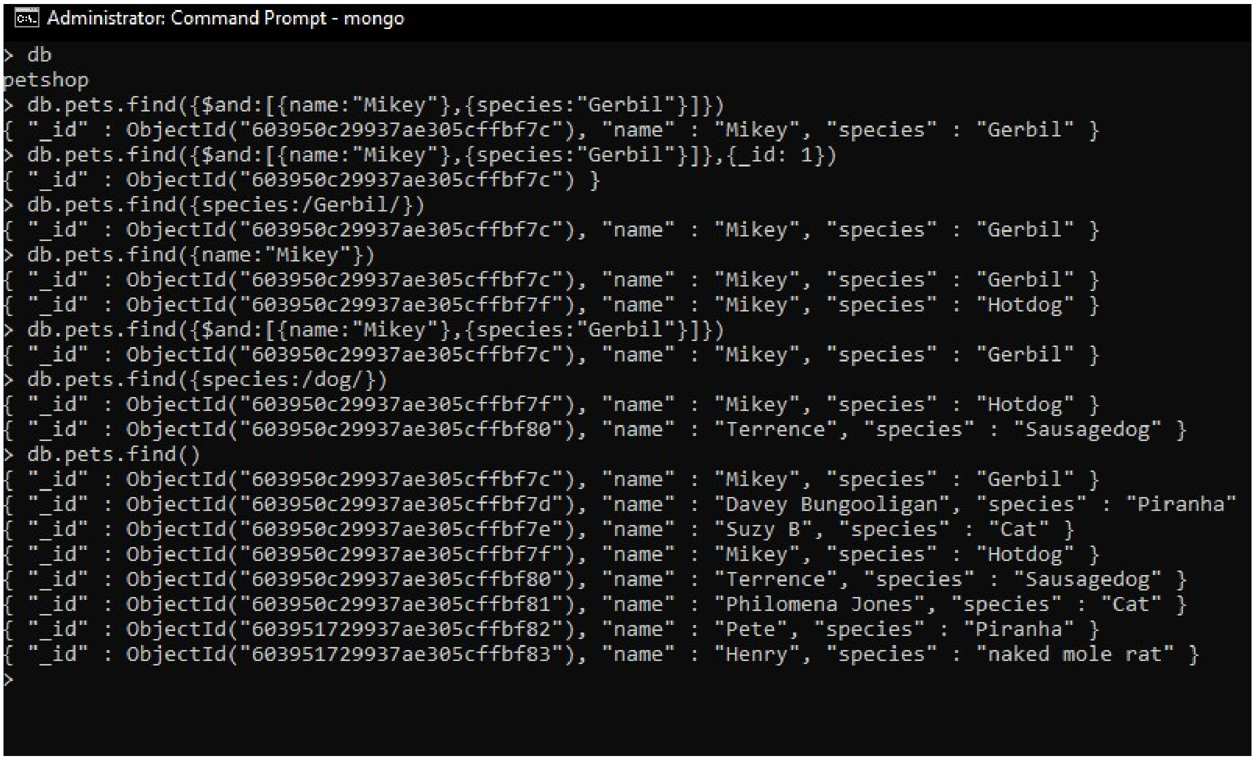


```
> db
Company
> db.customers.insert( [ {first_name:"Sam", last_name:"Smith"} , {first_name:"Jade", last_name:"Smith", gender:"female"}]);
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 2,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.customers.find().pretty();
{
        "_id" : ObjectId("6039455e9937ae305cffbf74"),
        "first_name" : "Jon",
        "last_name" : "Doe"
}
{
        "_id" : ObjectId("6039458a9937ae305cffbf75"),
        "first_name" : "John",
        "last_name" : "Smith"
}
{
        "_id" : ObjectId("603945bb9937ae305cffbf76"),
        "first_name" : "Alicia",
        "last_name" : "Zelaya"
}
{
        "_id" : ObjectId("603945dd9937ae305cffbf77"),
        "first_name" : "Jennifer",
        "last_name" : "Wallace"
}
{
        "_id" : ObjectId("603945f89937ae305cffbf78"),
        "first_name" : "Ahmad",
        "last_name" : "Jabbar"
}
{
        "_id" : ObjectId("603946179937ae305cffbf79"),
        "first_name" : "James",
        "last_name" : "Borg"
}
{
        "_id" : ObjectId("60394d659937ae305cffbf7a"),
        "first_name" : "Sam",
        "last_name" : "Smith"
}
{
        "_id" : ObjectId("60394d659937ae305cffbf7b"),
        "first_name" : "Jade",
        "last_name" : "Smith",
        "gender" : "female"
}
>
```

Exercise 9:



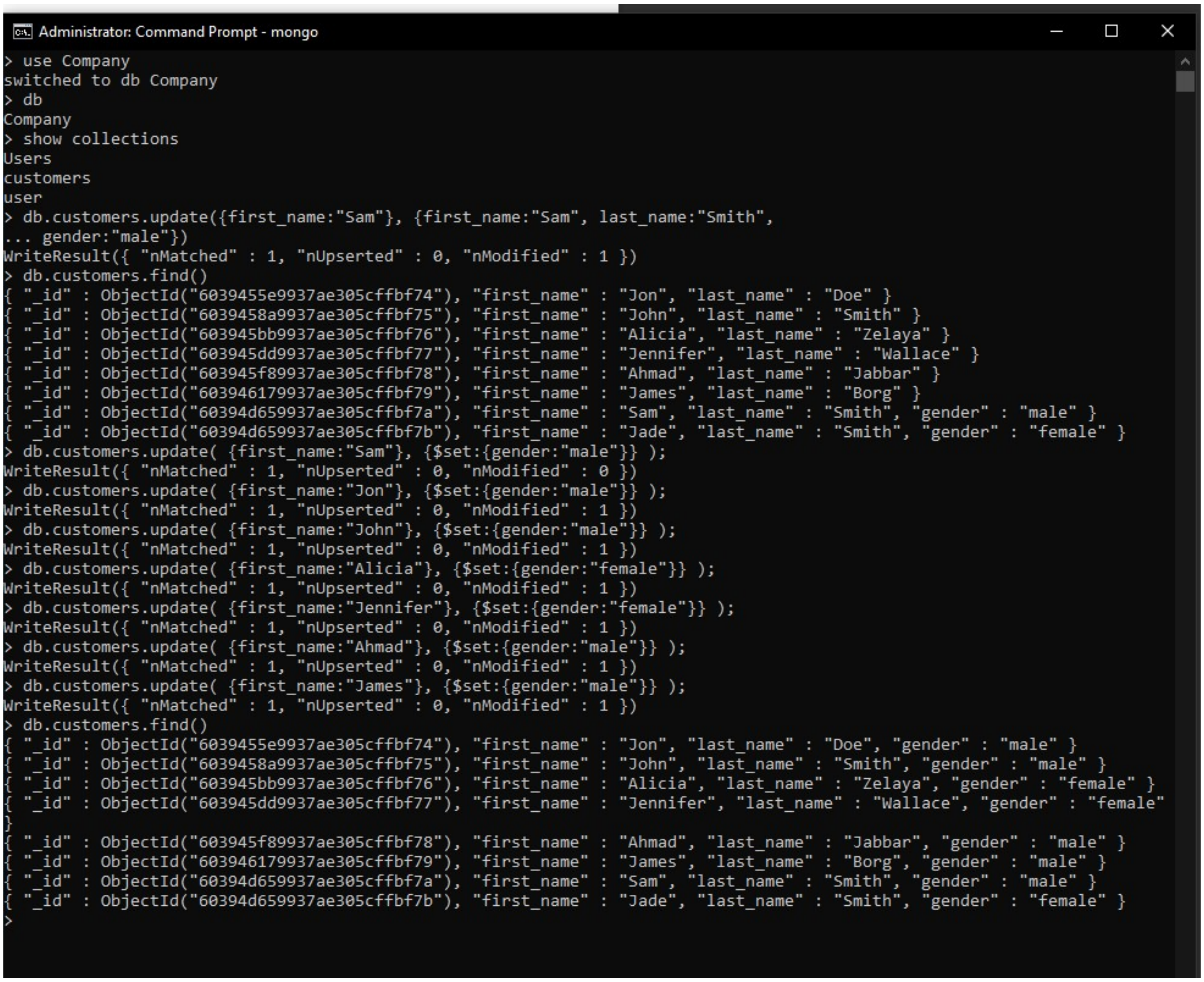
```
{ "ok" : 1 }
> db
petshop
> db.pets.insert([{name:"Mikey",species:"Gerbil"},{name:"Davey Bungooligan",species:"Piranha"},{name:"Suzy B",species:"C
at"},{name:"Mikey",species:"Hotdog"},{name:"Terrence",species:"Sausagedog"},{name:"Philomena Jones",species:"Cat"}])
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 6,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.pets.find()
{ "_id" : ObjectId("603950c29937ae305cffbf7c"), "name" : "Mikey", "species" : "Gerbil" }
{ "_id" : ObjectId("603950c29937ae305cffbf7d"), "name" : "Davey Bungooligan", "species" : "Piranha" }
{ "_id" : ObjectId("603950c29937ae305cffbf7e"), "name" : "Suzy B", "species" : "Cat" }
{ "_id" : ObjectId("603950c29937ae305cffbf7f"), "name" : "Mikey", "species" : "Hotdog" }
{ "_id" : ObjectId("603950c29937ae305cffbf80"), "name" : "Terrence", "species" : "Sausagedog" }
{ "_id" : ObjectId("603950c29937ae305cffbf81"), "name" : "Philomena Jones", "species" : "Cat" }
> db.pets.insert([{name:"Pete",species:"Piranha"},{name:"Henry",species:"naked mole rat"}])
BulkWriteResult({
        "writeErrors" : [ ],
        "writeConcernErrors" : [ ],
        "nInserted" : 2,
        "nUpserted" : 0,
        "nMatched" : 0,
        "nModified" : 0,
        "nRemoved" : 0,
        "upserted" : [ ]
})
> db.pets.find()
{ "_id" : ObjectId("603950c29937ae305cffbf7c"), "name" : "Mikey", "species" : "Gerbil" }
{ "_id" : ObjectId("603950c29937ae305cffbf7d"), "name" : "Davey Bungooligan", "species" : "Piranha" }
{ "_id" : ObjectId("603950c29937ae305cffbf7e"), "name" : "Suzy B", "species" : "Cat" }
{ "_id" : ObjectId("603950c29937ae305cffbf7f"), "name" : "Mikey", "species" : "Hotdog" }
{ "_id" : ObjectId("603950c29937ae305cffbf80"), "name" : "Terrence", "species" : "Sausagedog" }
{ "_id" : ObjectId("603950c29937ae305cffbf81"), "name" : "Philomena Jones", "species" : "Cat" }
{ "_id" : ObjectId("603951729937ae305cffbf82"), "name" : "Pete", "species" : "Piranha" }
{ "_id" : ObjectId("603951729937ae305cffbf83"), "name" : "Henry", "species" : "naked mole rat" }
```



```
> db
petshop
> db.pets.find({$and:[{name:"Mikey"},{species:"Gerbil"}]})
{ "_id" : ObjectId("603950c29937ae305cffbf7c"), "name" : "Mikey", "species" : "Gerbil" }
> db.pets.find({$and:[{name:"Mikey"},{species:"Gerbil"}]},{_id: 1})
{ "_id" : ObjectId("603950c29937ae305cffbf7c") }
> db.pets.find({species:/Gerbil/})
{ "_id" : ObjectId("603950c29937ae305cffbf7c"), "name" : "Mikey", "species" : "Gerbil" }
> db.pets.find({name:"Mikey"})
{ "_id" : ObjectId("603950c29937ae305cffbf7c"), "name" : "Mikey", "species" : "Gerbil" }
{ "_id" : ObjectId("603950c29937ae305cffbf7f"), "name" : "Mikey", "species" : "Hotdog" }
> db.pets.find({$and:[{name:"Mikey"},{species:"Gerbil"}]})
{ "_id" : ObjectId("603950c29937ae305cffbf7c"), "name" : "Mikey", "species" : "Gerbil" }
> db.pets.find({species:/dog/})
{ "_id" : ObjectId("603950c29937ae305cffbf7f"), "name" : "Mikey", "species" : "Hotdog" }
{ "_id" : ObjectId("603950c29937ae305cffbf80"), "name" : "Terrence", "species" : "Sausagedog" }
> db.pets.find()
{ "_id" : ObjectId("603950c29937ae305cffbf7c"), "name" : "Mikey", "species" : "Gerbil" }
{ "_id" : ObjectId("603950c29937ae305cffbf7d"), "name" : "Davey Bungooligan", "species" : "Piranha"
{ "_id" : ObjectId("603950c29937ae305cffbf7e"), "name" : "Suzy B", "species" : "Cat" }
{ "_id" : ObjectId("603950c29937ae305cffbf7f"), "name" : "Mikey", "species" : "Hotdog" }
{ "_id" : ObjectId("603950c29937ae305cffbf80"), "name" : "Terrence", "species" : "Sausagedog" }
{ "_id" : ObjectId("603950c29937ae305cffbf81"), "name" : "Philomena Jones", "species" : "Cat" }
{ "_id" : ObjectId("603951729937ae305cffbf82"), "name" : "Pete", "species" : "Piranha" }
{ "_id" : ObjectId("603951729937ae305cffbf83"), "name" : "Henry", "species" : "naked mole rat" }
>
```

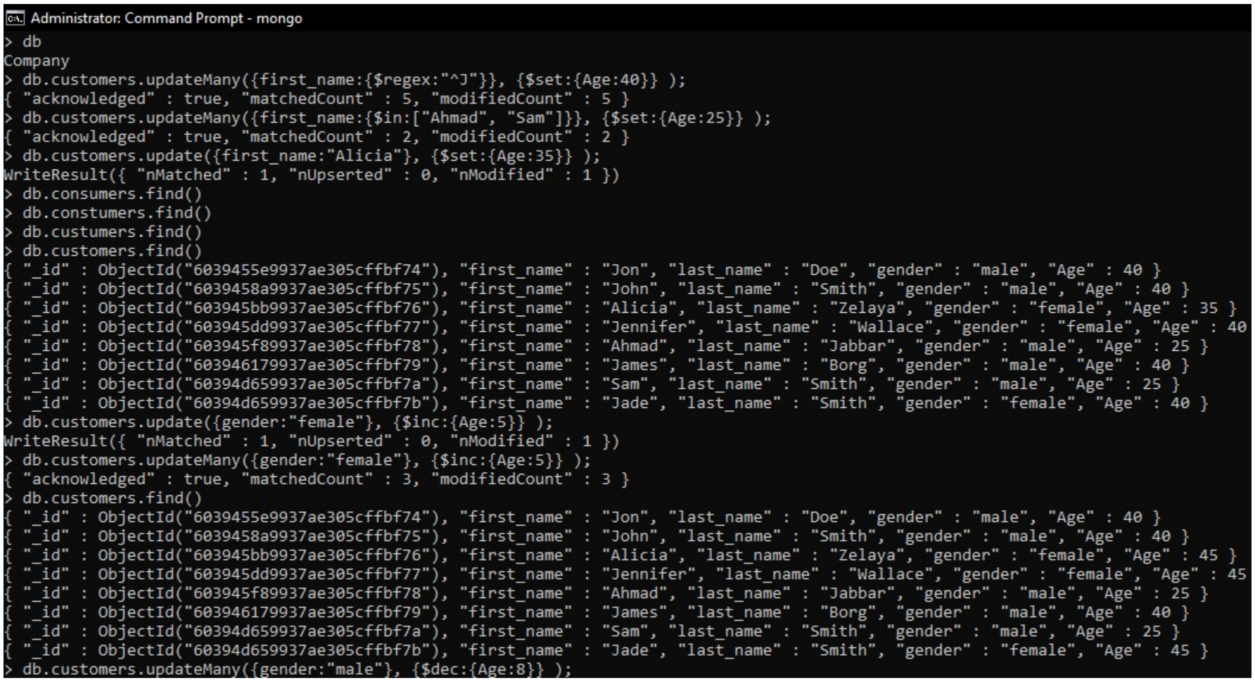
Exercise 10:

Adding gender column to all rows



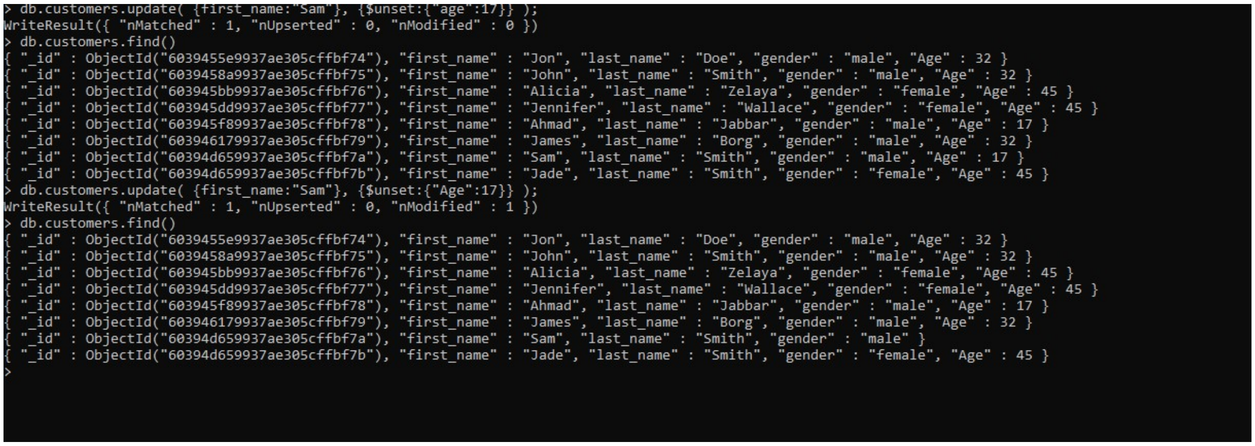
```
> use Company
switched to db Company
> db
Company
> show collections
Users
customers
user
> db.customers.update({first_name:"Sam"}, {first_name:"Sam", last_name:"Smith",
... gender:"male"})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe" }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith" }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya" }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace" }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar" }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg" }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male" }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female" }
> db.customers.update( {first_name:"Sam"}, {$set:{gender:"male"}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.customers.update( {first_name:"Jon"}, {$set:{gender:"male"}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.update( {first_name:"John"}, {$set:{gender:"male"}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.update( {first_name:"Alicia"}, {$set:{gender:"female"}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.update( {first_name:"Jennifer"}, {$set:{gender:"female"}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.update( {first_name:"Ahmad"}, {$set:{gender:"male"}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.update( {first_name:"James"}, {$set:{gender:"male"}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male" }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male" }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female" }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female"
}
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male" }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male" }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male" }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female" }
>
```

Adding and increment rows



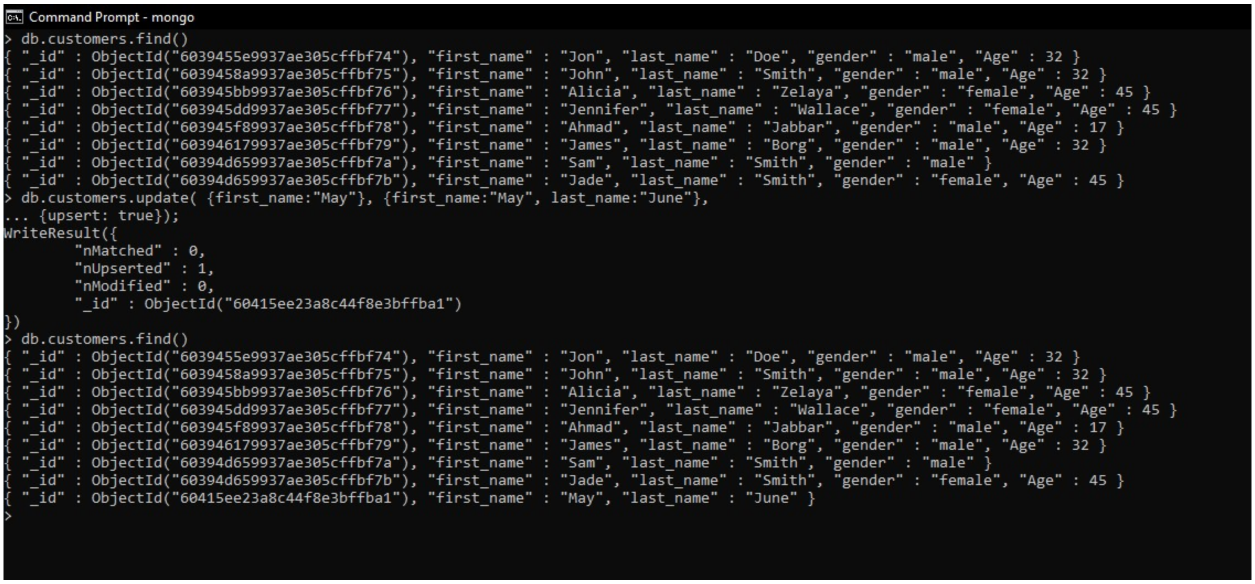
```
> db
Company
> db.customers.updateMany({first_name:{$regex:"^J"}}, {$set:{Age:40}} );
{ "acknowledged" : true, "matchedCount" : 5, "modifiedCount" : 5 }
> db.customers.updateMany({first_name:{$in:["Ahmad", "Sam"]}}, {$set:{Age:25}} );
{ "acknowledged" : true, "matchedCount" : 2, "modifiedCount" : 2 }
> db.customers.update({first_name:"Alicia"}, {$set:{Age:35}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.consumers.find()
> db.constumers.find()
> db.custumers.find()
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male", "Age" : 40 }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male", "Age" : 40 }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female", "Age" : 35 }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female", "Age" : 40
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male", "Age" : 25 }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male", "Age" : 40 }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male", "Age" : 25 }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female", "Age" : 40 }
> db.customers.update({gender:"female"}, {$inc:{Age:5}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.updateMany({gender:"female"}, {$inc:{Age:5}} );
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male", "Age" : 40 }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male", "Age" : 40 }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female", "Age" : 45
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male", "Age" : 25 }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male", "Age" : 40 }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male", "Age" : 25 }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female", "Age" : 45 }
> db.customers.updateMany({gender:"male"}, {$dec:{Age:8}} );
```

Unset columns



```
> db.customers.update( {first_name:"Sam"}, {$unset:{"age":17}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male", "Age" : 17 }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male", "Age" : 17 }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female", "Age" : 45 }
> db.customers.update( {first_name:"Sam"}, {$unset:{"Age":17}} );
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male", "Age" : 17 }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male" }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female", "Age" : 45 }
>
```

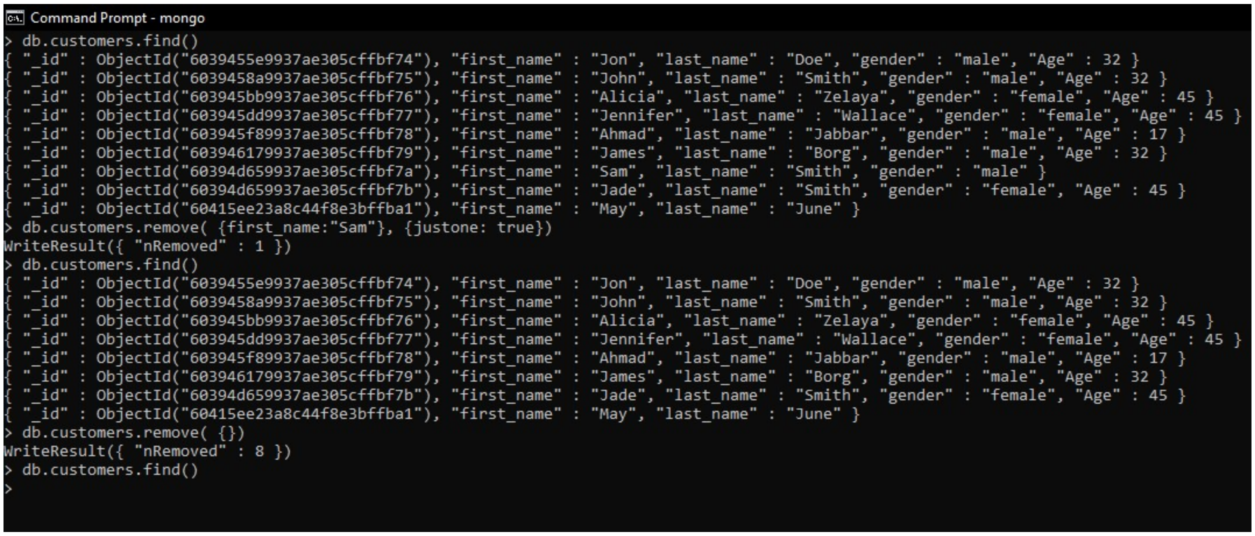
Upsert



```
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male", "Age" : 17 }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male" }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female", "Age" : 45 }
> db.customers.update( {first_name:"May"}, {first_name:"May", last_name:"June"},
... {upsert: true});
WriteResult({
        "nMatched" : 0,
        "nUpserted" : 1,
        "nModified" : 0,
        "_id" : ObjectId("60415ee23a8c44f8e3bffba1")
})
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male", "Age" : 17 }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male" }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("60415ee23a8c44f8e3bffba1"), "first_name" : "May", "last_name" : "June" }
>
```

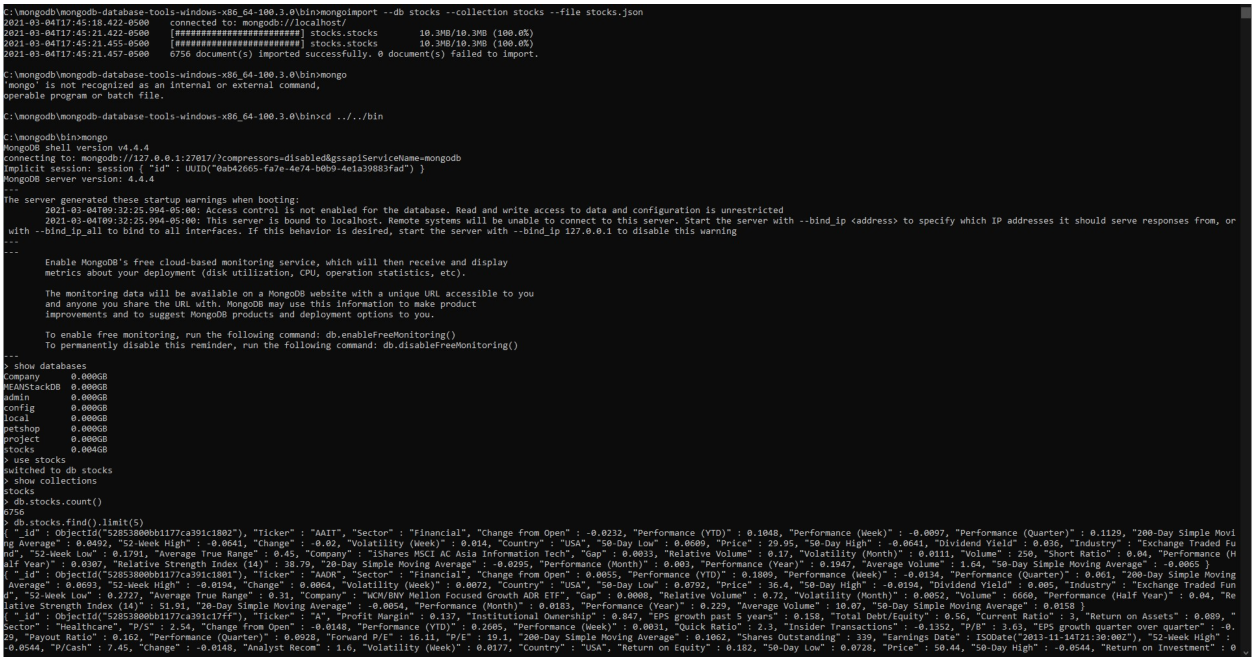
Exercise 11

Remove one and all columns



```
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male", "Age" : 17 }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("60394d659937ae305cffbf7a"), "first_name" : "Sam", "last_name" : "Smith", "gender" : "male" }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("60415ee23a8c44f8e3bffba1"), "first_name" : "May", "last_name" : "June" }
> db.customers.remove( {first_name:"Sam"}, {justone: true})
WriteResult({ "nRemoved" : 1 })
> db.customers.find()
{ "_id" : ObjectId("6039455e9937ae305cffbf74"), "first_name" : "Jon", "last_name" : "Doe", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("6039458a9937ae305cffbf75"), "first_name" : "John", "last_name" : "Smith", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("603945bb9937ae305cffbf76"), "first_name" : "Alicia", "last_name" : "Zelaya", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945dd9937ae305cffbf77"), "first_name" : "Jennifer", "last_name" : "Wallace", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("603945f89937ae305cffbf78"), "first_name" : "Ahmad", "last_name" : "Jabbar", "gender" : "male", "Age" : 17 }
{ "_id" : ObjectId("603946179937ae305cffbf79"), "first_name" : "James", "last_name" : "Borg", "gender" : "male", "Age" : 32 }
{ "_id" : ObjectId("60394d659937ae305cffbf7b"), "first_name" : "Jade", "last_name" : "Smith", "gender" : "female", "Age" : 45 }
{ "_id" : ObjectId("60415ee23a8c44f8e3bffba1"), "first_name" : "May", "last_name" : "June" }
> db.customers.remove( {})
WriteResult({ "nRemoved" : 8 })
> db.customers.find()
>
```

Exercise 12



```
C:\mongodb\mongodb-database-tools-windows-x86_64-100.3.0\bin>mongoimport --db stocks --collection stocks --file stocks.json
2021-03-04T17:45:18.422-0500    connected to: mongodb://localhost/
2021-03-04T17:45:21.422-0500    [########################] stocks.stocks        10.3MB/10.3MB (100.0%)
2021-03-04T17:45:21.455-0500    [########################] stocks.stocks        10.3MB/10.3MB (100.0%)
2021-03-04T17:45:21.457-0500    6756 document(s) imported successfully. 0 document(s) failed to import.

C:\mongodb\mongodb-database-tools-windows-x86_64-100.3.0\bin>mongo
'mongo' is not recognized as an internal or external command,
operable program or batch file.

C:\mongodb\mongodb-database-tools-windows-x86_64-100.3.0\bin>cd ../../bin

C:\mongodb\bin>mongo
MongoDB shell version v4.4.4
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0ab42665-fa7e-4e74-b0b9-4e1a39883fad") }
MongoDB server version: 4.4.4
---
The server generated these startup warnings when booting:
        2021-03-04T09:32:25.994-05:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
        2021-03-04T09:32:25.994-05:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or
 with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
---
---
        Enable MongoDB's free cloud-based monitoring service, which will then receive and display
        metrics about your deployment (disk utilization, CPU, operation statistics, etc).

        The monitoring data will be available on a MongoDB website with a unique URL accessible to you
        and anyone you share the URL with. MongoDB may use this information to make product
        improvements and to suggest MongoDB products and deployment options to you.

        To enable free monitoring, run the following command: db.enableFreeMonitoring()
        To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
> show databases
Company      0.000GB
MEANStackDB  0.000GB
admin        0.000GB
config       0.000GB
local        0.000GB
petshop      0.000GB
project      0.000GB
stocks       0.004GB
> use stocks
switched to db stocks
> show collections
stocks
> db.stocks.count()
6756
> db.stocks.find().limit(5)
{ "_id" : ObjectId("52853800bb1177ca391c1802"), "Ticker" : "AAIT", "Sector" : "Financial", "Change from Open" : -0.0232, "Performance (YTD)" : 0.1048, "Performance (Week)" : -0.0097, "Performance (Quarter)" : 0.1129, "200-Day Simple Movi
ng Average" : 0.0492, "52-Week High" : -0.0641, "Change" : -0.02, "Volatility (Week)" : 0.014, "Country" : "USA", "50-Day Low" : 0.0609, "Price" : 29.95, "50-Day High" : -0.0641, "Dividend Yield" : 0.036, "Industry" : "Exchange Traded Fu
nd", "52-Week Low" : 0.1791, "Average True Range" : 0.45, "Company" : "iShares MSCI AC Asia Information Tech", "Gap" : 0.0033, "Relative Volume" : 0.17, "Volatility (Month)" : 0.0111, "Volume" : 250, "Short Ratio" : 0.04, "Performance (H
alf Year)" : 0.0307, "Relative Strength Index (14)" : 38.79, "20-Day Simple Moving Average" : -0.0295, "Performance (Month)" : 0.003, "Performance (Year)" : 0.1947, "Average Volume" : 1.64, "50-Day Simple Moving Average" : -0.0065 }
{ "_id" : ObjectId("52853800bb1177ca391c1801"), "Ticker" : "AADR", "Sector" : "Financial", "Change from Open" : 0.0055, "Performance (YTD)" : 0.1809, "Performance (Week)" : -0.0134, "Performance (Quarter)" : 0.061, "200-Day Simple Moving
 Average" : 0.0693, "52-Week High" : -0.0194, "Change" : 0.0064, "Volatility (Week)" : 0.0072, "Country" : "USA", "50-Day Low" : 0.0792, "Price" : 36.4, "50-Day High" : -0.0194, "Dividend Yield" : 0.005, "Industry" : "Exchange Traded Fun
d", "52-Week Low" : 0.2727, "Average True Range" : 0.31, "Company" : "WCM/BNY Mellon Focused Growth ADR ETF", "Gap" : 0.0008, "Relative Volume" : 0.72, "Volatility (Month)" : 0.0052, "Volume" : 6660, "Performance (Half Year)" : 0.04, "Re
lative Strength Index (14)" : 51.91, "20-Day Simple Moving Average" : -0.0054, "Performance (Month)" : 0.0183, "Performance (Year)" : 0.229, "Average Volume" : 10.07, "50-Day Simple Moving Average" : 0.0158 }
{ "_id" : ObjectId("52853800bb1177ca391c17ff"), "Ticker" : "A", "Profit Margin" : 0.137, "Institutional Ownership" : 0.847, "EPS growth past 5 years" : 0.158, "Total Debt/Equity" : 0.56, "Current Ratio" : 3, "Return on Assets" : 0.089, "
Sector" : "Healthcare", "P/S" : 2.54, "Change from Open" : -0.0148, "Performance (YTD)" : 0.2605, "Performance (Week)" : 0.0031, "Quick Ratio" : 2.3, "Insider Transactions" : -0.1352, "P/B" : 3.63, "EPS growth quarter over quarter" : -0.
29, "Payout Ratio" : 0.162, "Performance (Quarter)" : 0.0928, "Forward P/E" : 16.11, "P/E" : 19.1, "200-Day Simple Moving Average" : 0.1062, "Shares Outstanding" : 339, "Earnings Date" : ISODate("2013-11-14T21:30:00Z"), "52-Week High" :
-0.0544, "P/Cash" : 7.45, "Change" : -0.0148, "Analyst Recom" : 1.6, "Volatility (Week)" : 0.0177, "Country" : "USA", "Return on Equity" : 0.182, "50-Day Low" : 0.0728, "Price" : 50.44, "50-Day High" : -0.0544, "Return on Investment" : 0
```