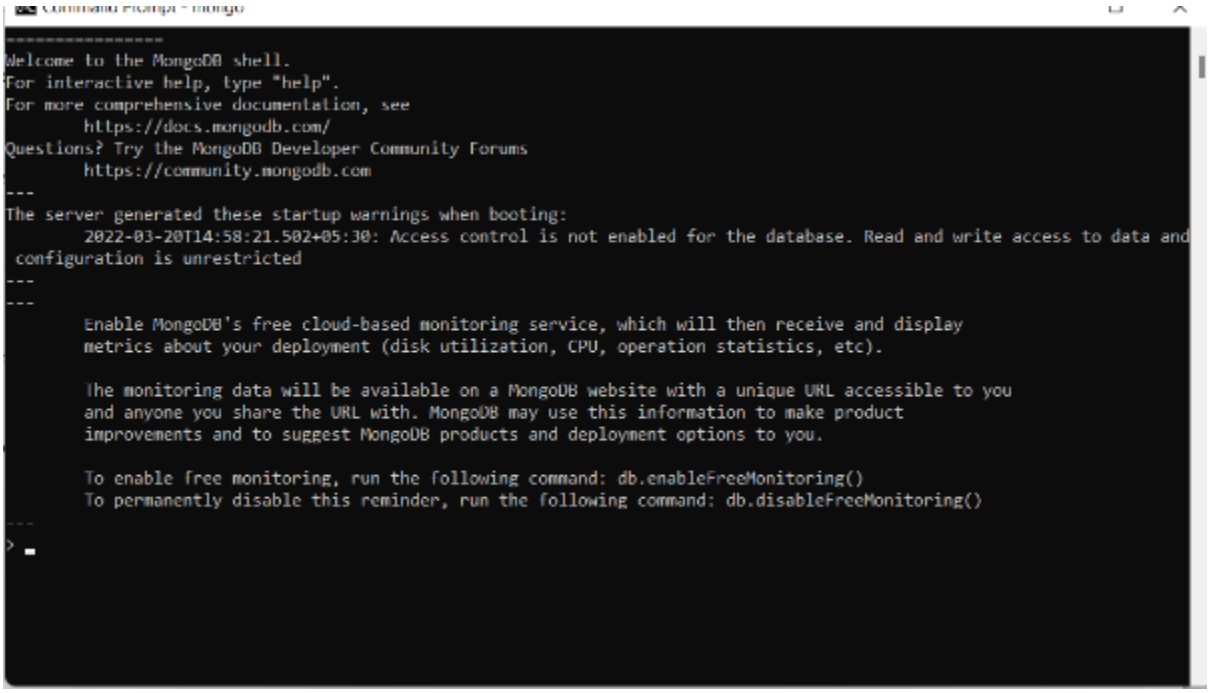
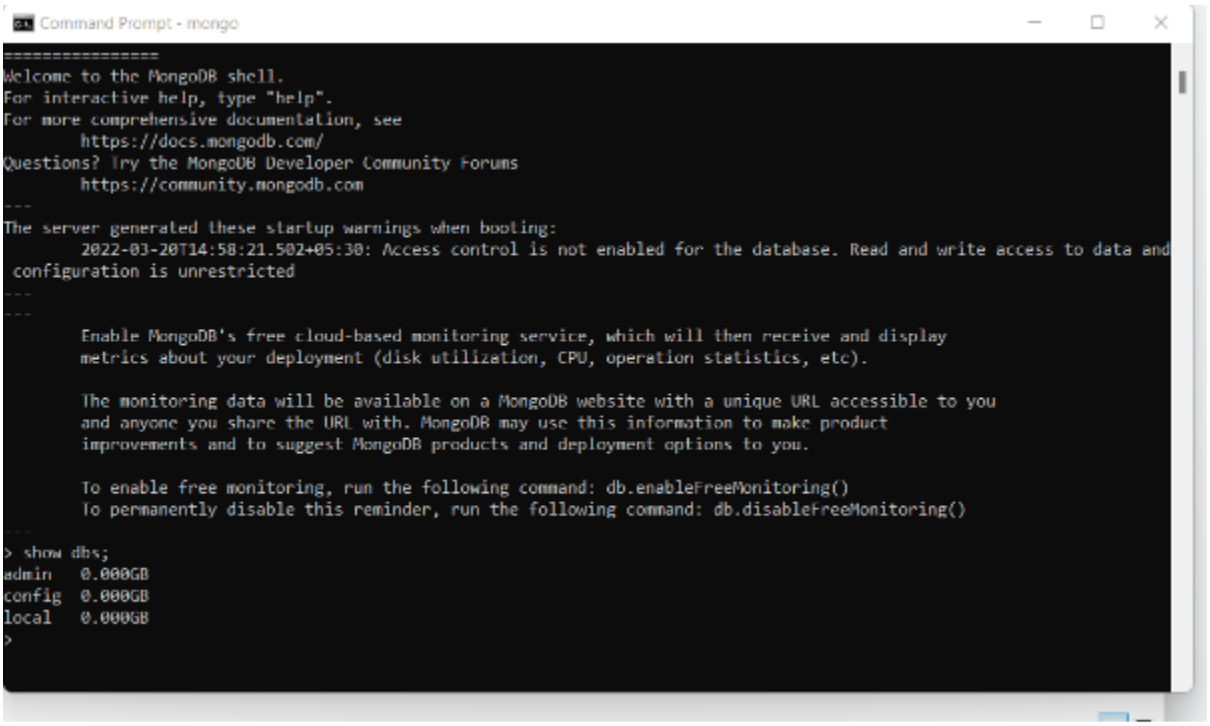
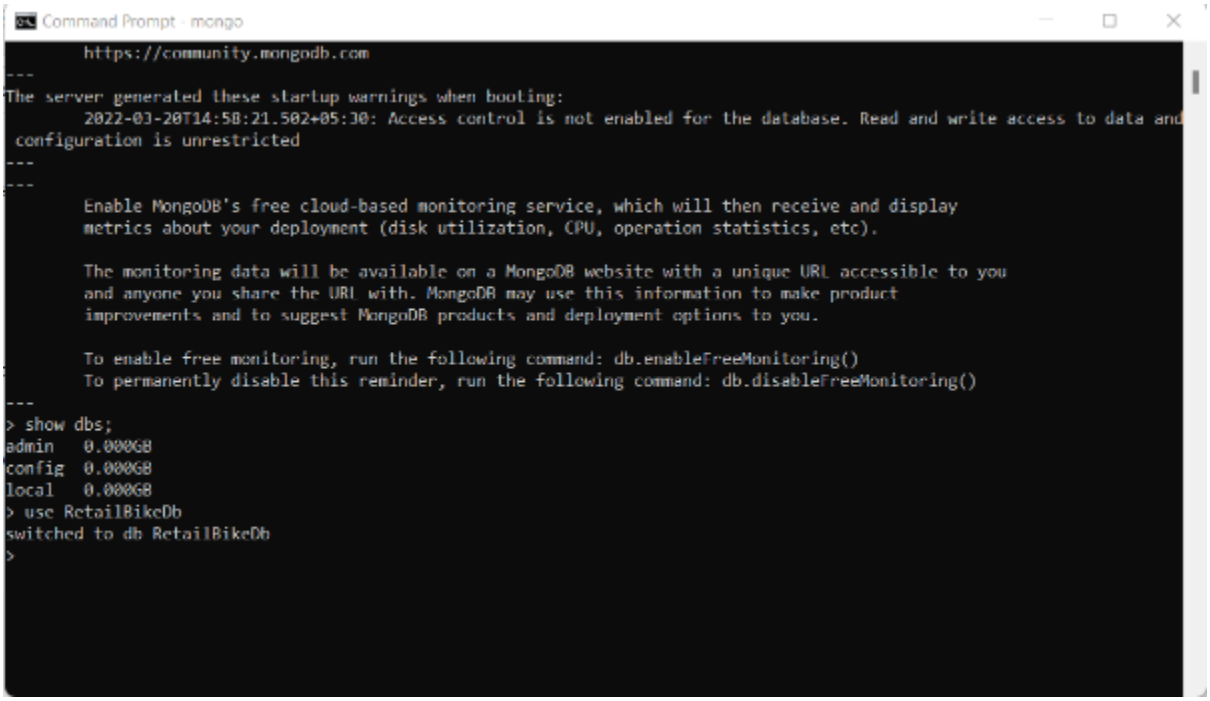
1)First start the mongo Db using command **mongo**



show dbs;

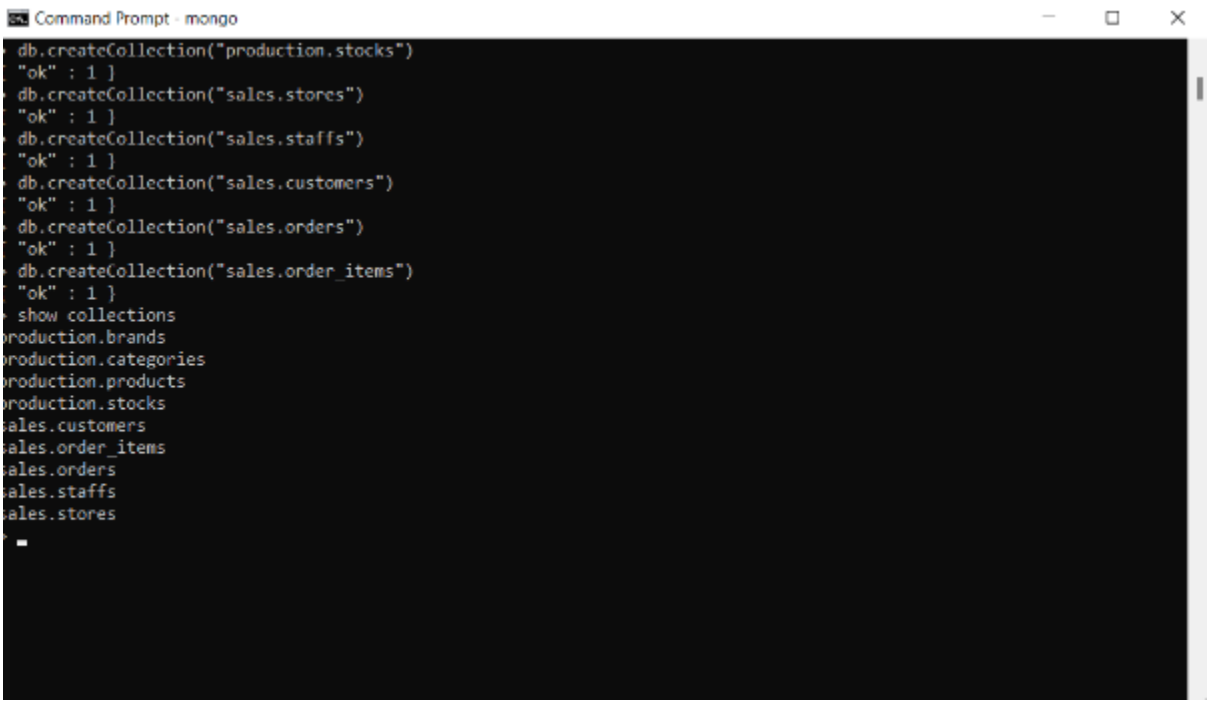


2) use RetailBikeDb



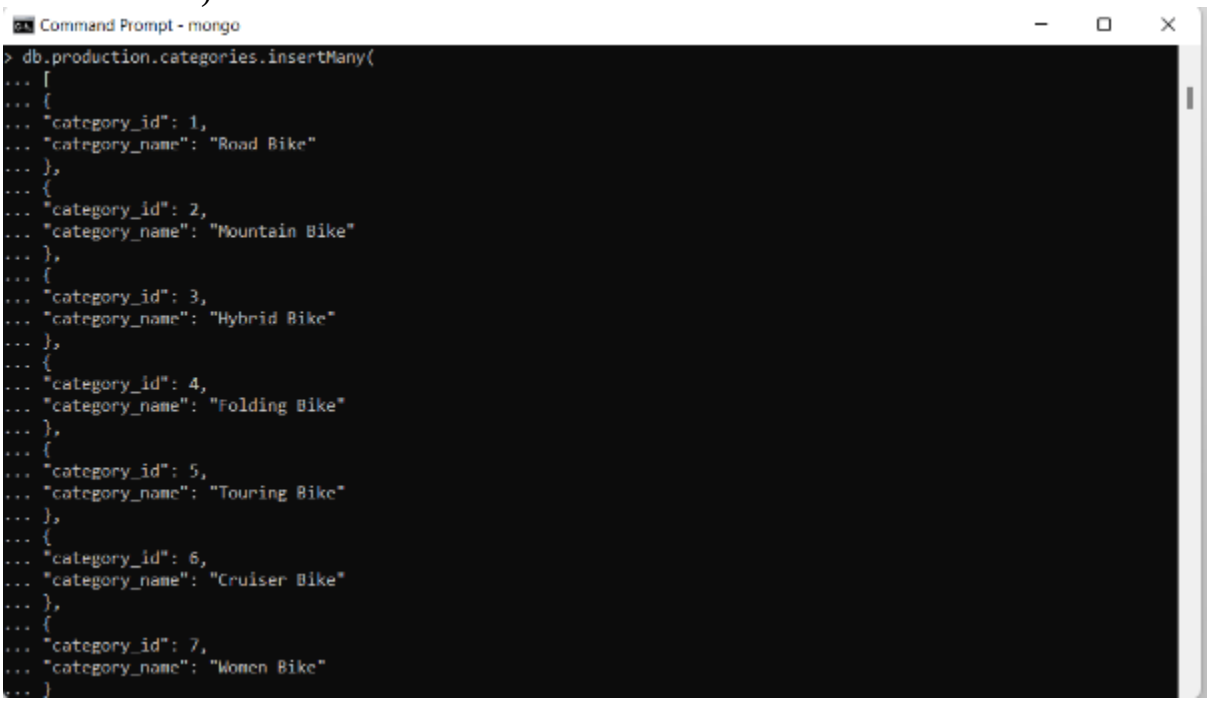
3) Creating and showing collections in MongoDB

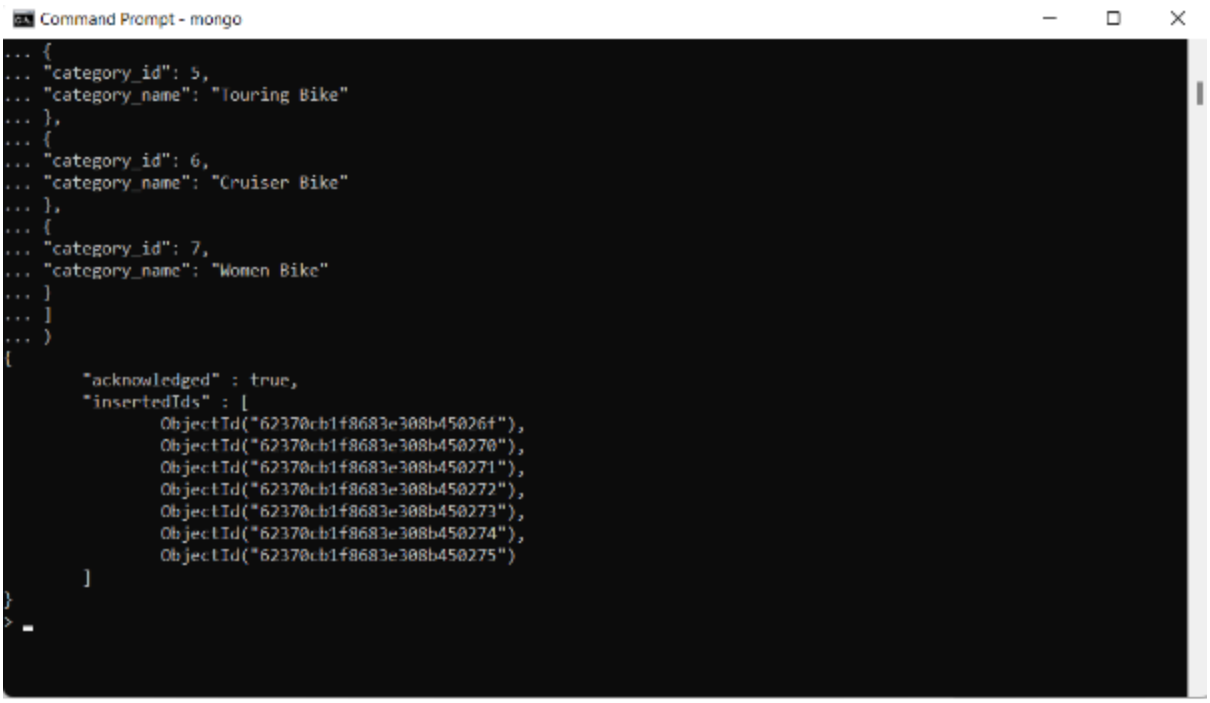
4) We can show list of collection using “Show collections” commands in MongoDB



1. Create Operations
   - db.collection.insertOne()
   - db.collection.insertMany()
   1. production.categories
      db.production.categories.insertMany(
      [
      {
      "category_id": 1,
      "category_name": "Road Bike"
      },
      {
      "category_id": 2,
      "category_name": "Mountain Bike"
      },
      {
      "category_id": 3,
      "category_name": "Hybrid Bike"
      },

```
{
"category_id": 4,
"category_name": "Folding Bike"
},
{
"category_id": 5,
"category_name": "Touring Bike"
},
{
"category_id": 6,
"category_name": "Cruiser Bike"
},
{
"category_id": 7,
"category_name": "Women Bike"
}
]
)
```

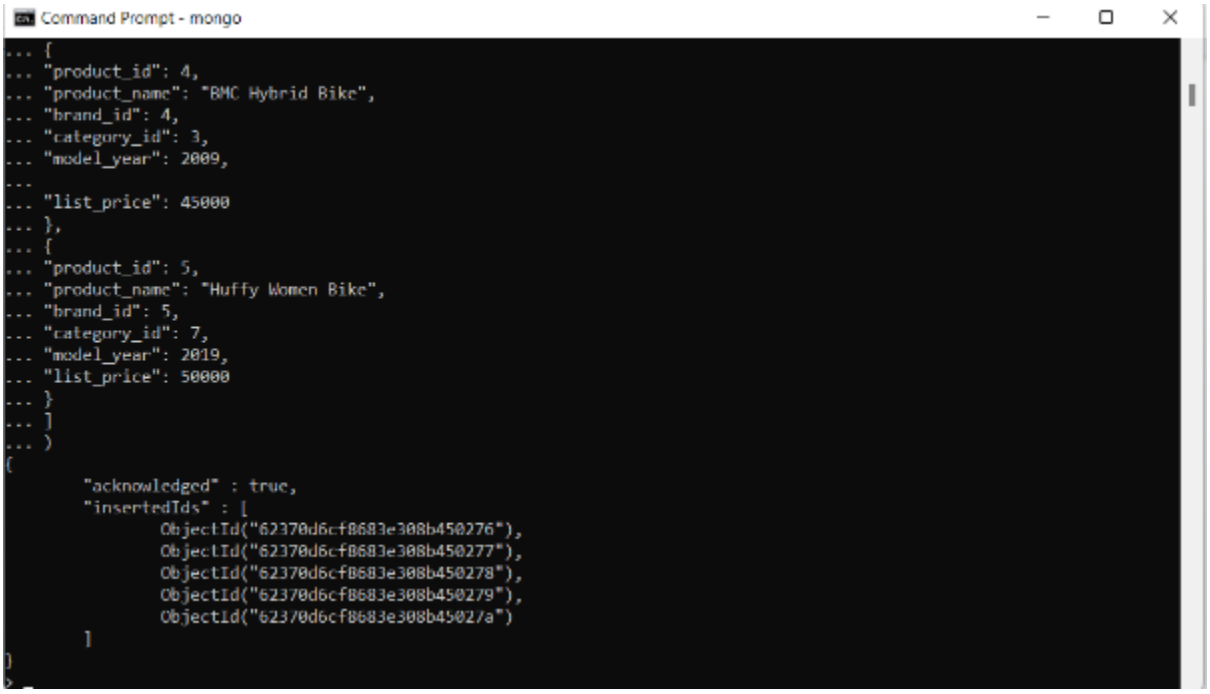




2. production.products

```
db.production.products.insertMany(
[
{
"product_id": 1,
"product_name": "Honda Superfast",
"brand_id": 1,
"category_id": 1,
"model_year": 1994,
"list_price": 25000
},
{
"product_id": 2,
"product_name": "6KU Bikes",
"brand_id": 2,
"category_id": 4,
"model_year": 2000,
"list_price": 30000
},
{
"product_id": 3,
"product_name": "Bianchi",
"brand_id": 3,
"category_id": 2,
"model_year": 2002,
"list_price": 30000
},
```

```
{
"product_id": 4,
"product_name": "BMC Hybrid Bike",
"brand_id": 4,
"category_id": 3,
"model_year": 2009,

"list_price": 45000
},
{
"product_id": 5,
"product_name": "Huffy Women Bike",
"brand_id": 5,
"category_id": 7,
"model_year": 2019,
"list_price": 50000
}
]
)
```



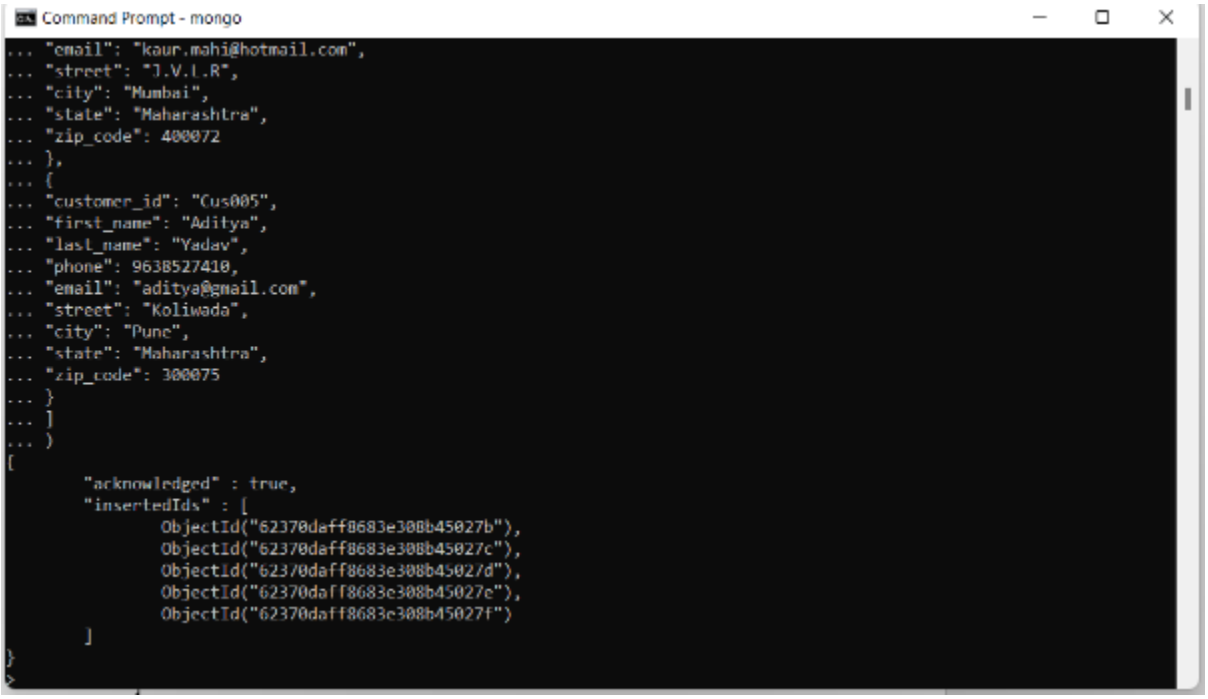
3. sales.customers

```
db.sales.customers.insertMany(
[
{
```

"customer_id": "Cus001",
"first_name": "Jay",
"last_name": "Mehta",
"phone": 1234567890,
"email": "jay@gmail.com",
"street": "T.P Road",
"city": "Mumbai",
"state": "Maharashtra",
"zip_code": 400072
},
{
"customer_id": "Cus002",
"first_name": "Ruhi",
"last_name": "Singh",
"phone": 7894561230,
"email": "ruhi@yahoo.com",
"street": "M.G Chauk",
"city": "Mumbai",
"state": "Maharashtra",
"zip_code": 400072
},
{
"customer_id": "Cus003",
"first_name": "Aria",
"last_name": "Josh",
"phone": 1245789630,
"email": "aria.josh@gmail.com",
"street": "JVM",
"city": "Gandhi Nagar",

"state": "Gujrat",
"zip_code": 401235
},
{
"customer_id": "Cus004",
"first_name": "Mahi",
"last_name": "Kaur",
"phone": 4567890123,
"email": "kaur.mahi@hotmail.com",
"street": "J.V.L.R",
"city": "Mumbai",
"state": "Maharashtra",
"zip_code": 400072
},
{
"customer_id": "Cus005",

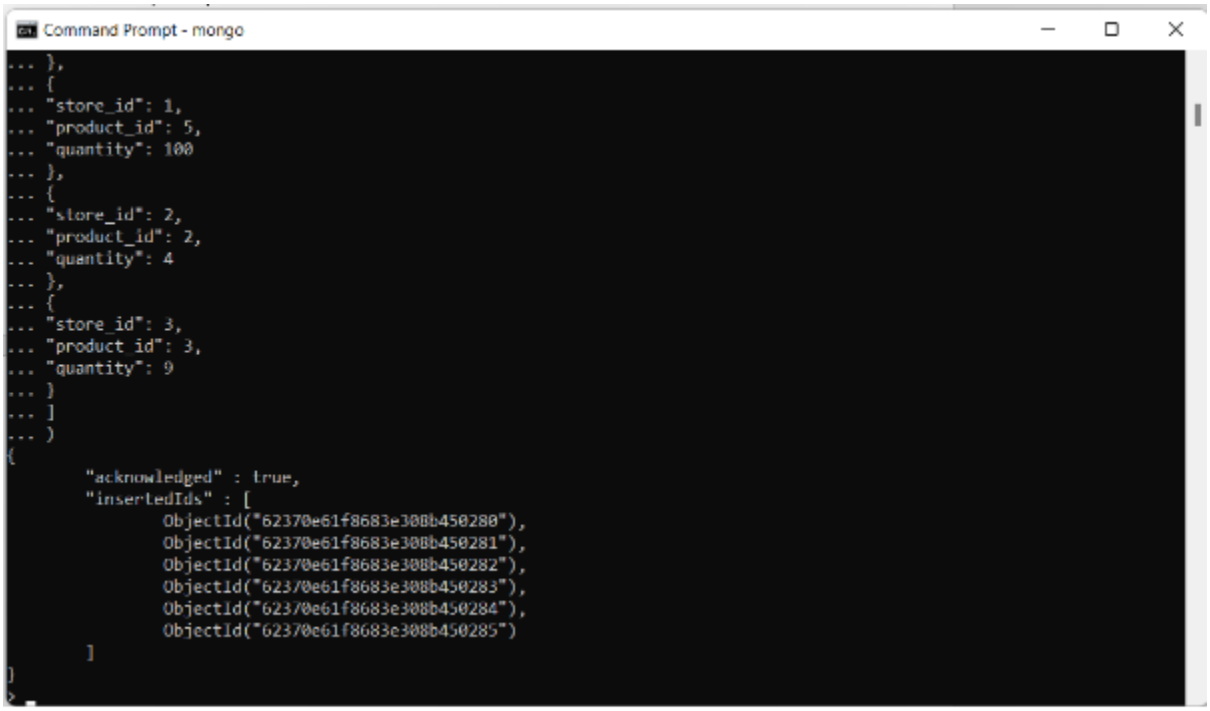
"first_name": "Aditya",
"last_name": "Yadav",
"phone": 9638527410,
"email": "aditya@gmail.com",
"street": "Koliwada",
"city": "Pune",
"state": "Maharashtra",
"zip_code": 300075
}
]
)



4.production.stocks

db.production.stocks.insertMany(
[
{
"store_id": 1,
"product_id": 1,
"quantity": 15
},
{
"store_id": 2,
"product_id": 4,
"quantity": 20
},
{
"store_id": 3,

```
"product_id": 5,
"quantity": 30
},
{
"store_id": 1,
"product_id": 5,
"quantity": 100
},
{
"store_id": 2,
"product_id": 2,
"quantity": 4
},
{
"store_id": 3,
"product_id": 3,
"quantity": 9
}
]
)
```



5.sales.order_items

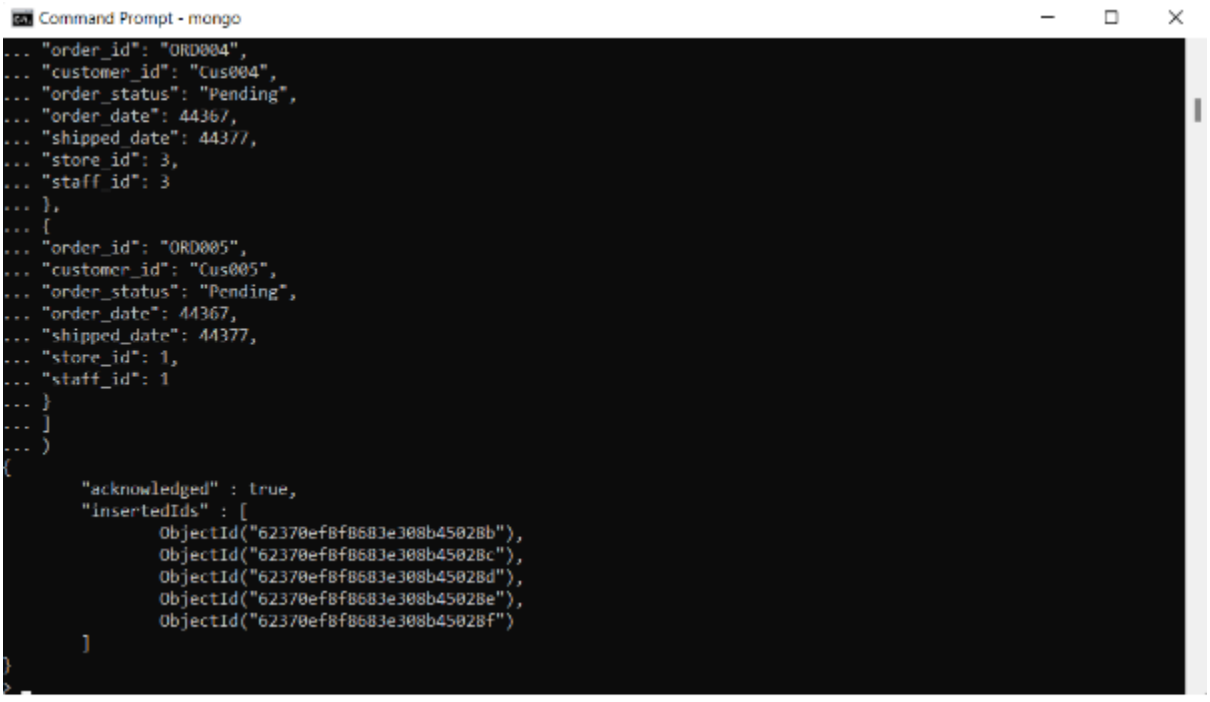
```
db.sales.order_items.insertMany(
```

```
[
{
"order_id": "ORD001",
"product_id": 1,
"quantity": 2,
"list_price": 50000
},
{
"order_id": "ORD002",
"product_id": 2,
"quantity": 3,
"list_price": 90000
},
{
"order_id": "ORD003",
"product_id": 3,
"quantity": 1,
"list_price": 30000
},
{
"order_id": "ORD004",
"product_id": 4,
"quantity": 8,
"list_price": 360000
},
{
"order_id": "ORD005",
"product_id": 5,
"quantity": 2,
"list_price": 100000
}

]
)
```

6.sales.orders

```
db.sales.orders.insertMany(
[
{
"order_id": "ORD001",
"customer_id": "Cus001",
"order_status": "Completed",
"order_date": 43992,
"shipped_date": 43994,
"store_id": 1,
"staff_id": 1
},
{
"order_id": "ORD002",
"customer_id": "Cus002",
"order_status": "Completed",
"order_date": 44221,
"shipped_date": 44227,
"store_id": 2,
"staff_id": 2
},
{
"order_id": "ORD003",
"customer_id": "Cus003",
```

```
"order_status": "Completed",
"order_date": 44306,
"shipped_date": 44314,
"store_id": 2,
"staff_id": 2

},
{
"order_id": "ORD004",
"customer_id": "Cus004",
"order_status": "Pending",
"order_date": 44367,
"shipped_date": 44377,
"store_id": 3,
"staff_id": 3
},
{
"order_id": "ORD005",
"customer_id": "Cus005",
"order_status": "Pending",
"order_date": 44367,
"shipped_date": 44377,
"store_id": 1,
"staff_id": 1
}
]
)
```



7.sales.staffs

```
db.sales.staffs.insertMany(
[
{
"staff_id": 1,
"first_name": "Pushpa",
"last_name": "Yadav",
"email": "pushpa@gmail.com",
"phone": 9999999999,
"active": "Yes",
"store_id": 1,
"manager_id": 1
},
{
"staff_id": 2,
"first_name": "Sadiksha",
"last_name": "Singh",
"email": "sadiksha@gmail.com",
"phone": 8888888888,
"active": "Yes",
"store_id": 2,
"manager_id": 1
},
{
"staff_id": 3,
"first_name": "Priya",
"last_name": "Nadar",
"email": "priya@gmail.com",
"phone": 7777777777,
"active": "Yes",
"store_id": 3,
"manager_id": 1

}
]
)
```

8. sales.stores

```
db.sales.stores.insertMany(
[
{
"store_id": 1,
"store_name": "Ambika Showroom",
"phone": 123456,
"email": "ambika@gmail.com",
"street": "GP",
"city": "Mumbai",
"state": "Maharashtra",
"zip_code": 400072
},
{
"store_id": 2,
"store_name": "Yash Bikes",
"phone": 789456,
"email": "yash.bikes@yahoo.com",
"street": "H.G",
"city": "Pune",
"state": "Maharashtra",
"zip_code": 300075
},
{
"store_id": 3,
"store_name": "Josh Automobiles",
"phone": 456983,
```

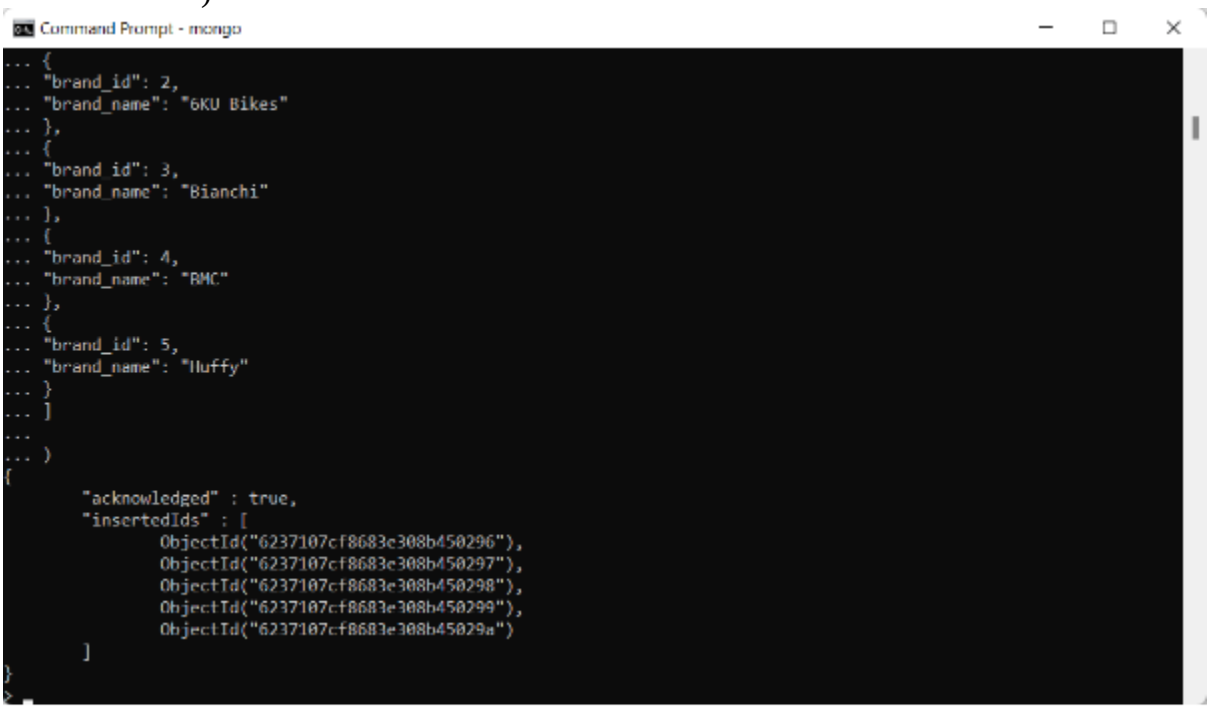
"email": "josh@gmail.com",

"street": "M.G",
"city": "Gandhi Nagar",
"state": "Gujrat",
"zip_code": 401235
}
]
)

9. production.brands

db.production.brands.insertMany(
[
{
"brand_id": 1,
"brand_name": "Honda"
},
{
"brand_id": 2,
"brand_name": "6KU Bikes"
},
{
"brand_id": 3,
"brand_name": "Bianchi"
},
{
"brand_id": 4,

```
"brand_name": "BMC"
},
{
"brand_id": 5,
"brand_name": "Huffy"
}
]

)
```



1. Read Operations
   MongoDB has two methods of reading documents from a collection:
   - db.collection.find()
   - db.collection.findOne()
   - find()

Example

Example: db.production.brands.find({"brand_name" : "Honda"})

- findOne()

Syntax- db.{collection}.findOne({query}, {projection})
Example : db.sales.order_items.findOne({"quantity": 2})

Query Documents

a. Specify Equality Condition

db.production.products.find() - to show all the documents

db.production.products.find({"list_price": 30000}) - to show only documents which have
"list_price" as 30000

b. Specify Conditions Using Query Operators

db.sales.customers.find() - to show all the documents
db.sales.customers.find({city: { $in: ["Mumbai", "Pune"] } }) - to show all the documents where
"city" is either "Mumbai" or "Pune"

C . Specify AND Conditions

db.sales.order_items.find() - to show all the documents
db.sales.order_items.find({quantity: 2, list_price : { $gt: 70000}}) - Here we are trying to find how
sales order item documents for which quantity is 2 and list price is greater than 70000

d. Specify OR Conditions

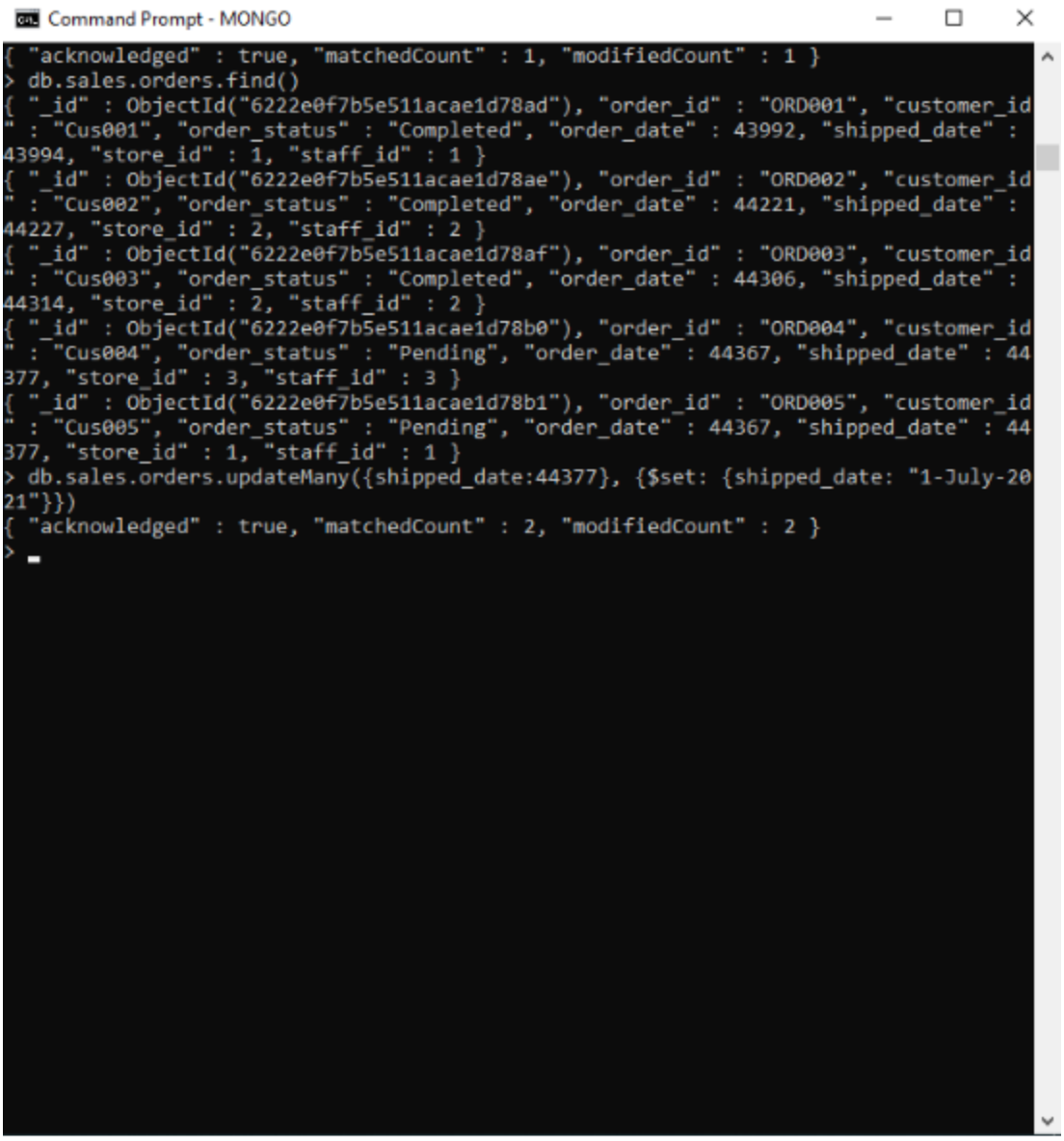
db.production.products.find() - to show all the documents

db.production.products.find({ $or: [{ product_name: "Honda Superfast" }, { model_year : { $lt: 2003 } }] }) - to show all the document which is either "product name" as "Honda Superfast" or "model year" is less than year 2003

3. Update Operations
   For MongoDB CRUD, there are three different methods of updating documents:
   - db.collection.updateOne()
   - db.collection.updateMany()
   - db.collection.replaceOne()

1. updateOne()
   db.sales.staffs.find() - to show currect document in the system
   db.sales.staffs.updateOne({first_name: "Pushpa"}, {$set:{phone: 999999988}}) - update mobile
   number from 9999999999 to 999999988 for document name where name is " first_name " in
   collection

1. updateMany()
   db.sales.orders.find() - to show current document in the system
   db.sales.orders.updateMany({shipped_date:44377}, {$set: {shipped_date: "1-July-2021"}}) – with
   the help of this command we are updating “shipped_date” to "1-July-2021" where shipped_date is
   44377



3.replaceOne()

4. Delete Operations

   MongoDB has two different methods of deleting records from a collection:

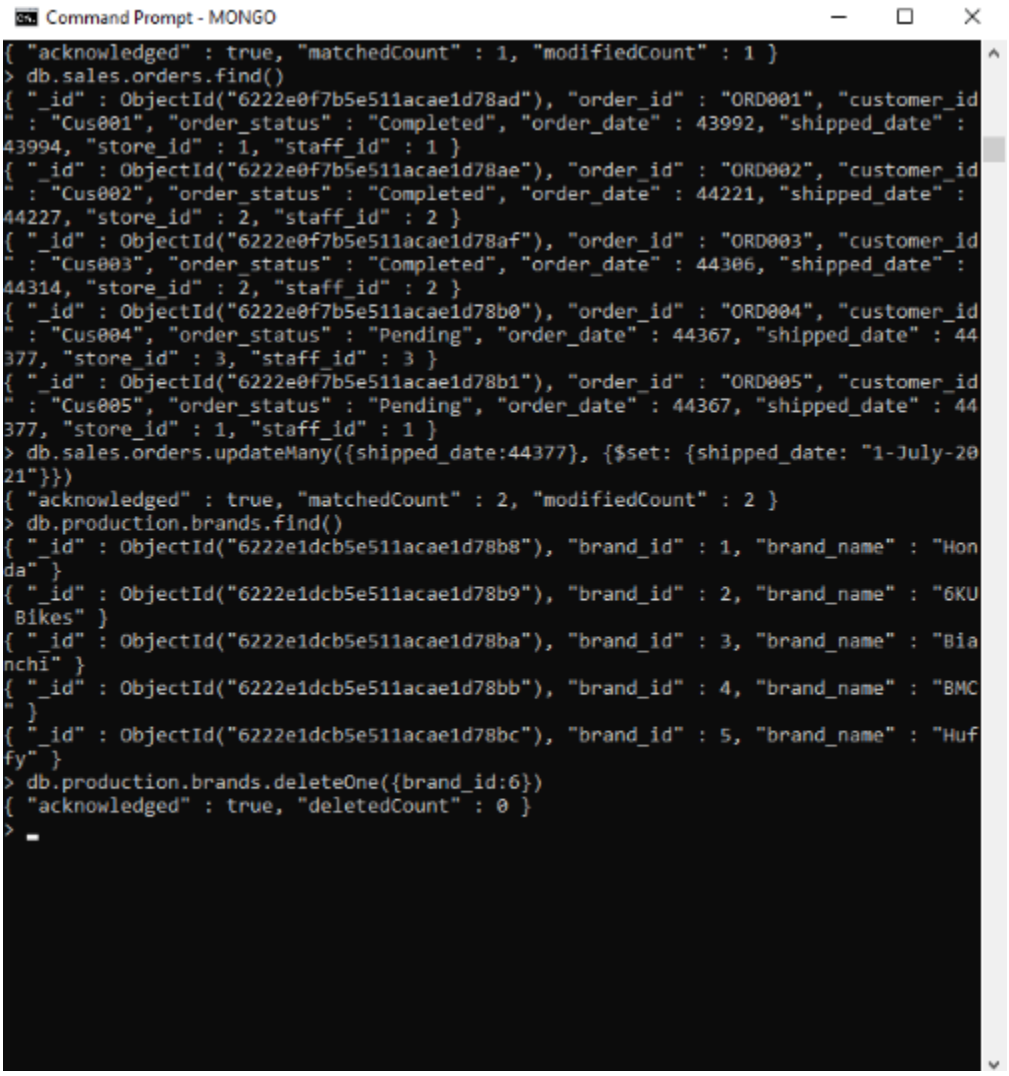
   - db.collection.delete004Fne()
   - db.collection.deleteMany()

     1. deleteOne()

        Example –

        db.production.brands.find() - to show current documents in collection

        db.production.brands.deleteOne({brand_id:6}) - delete the document where "brand_id" is 6

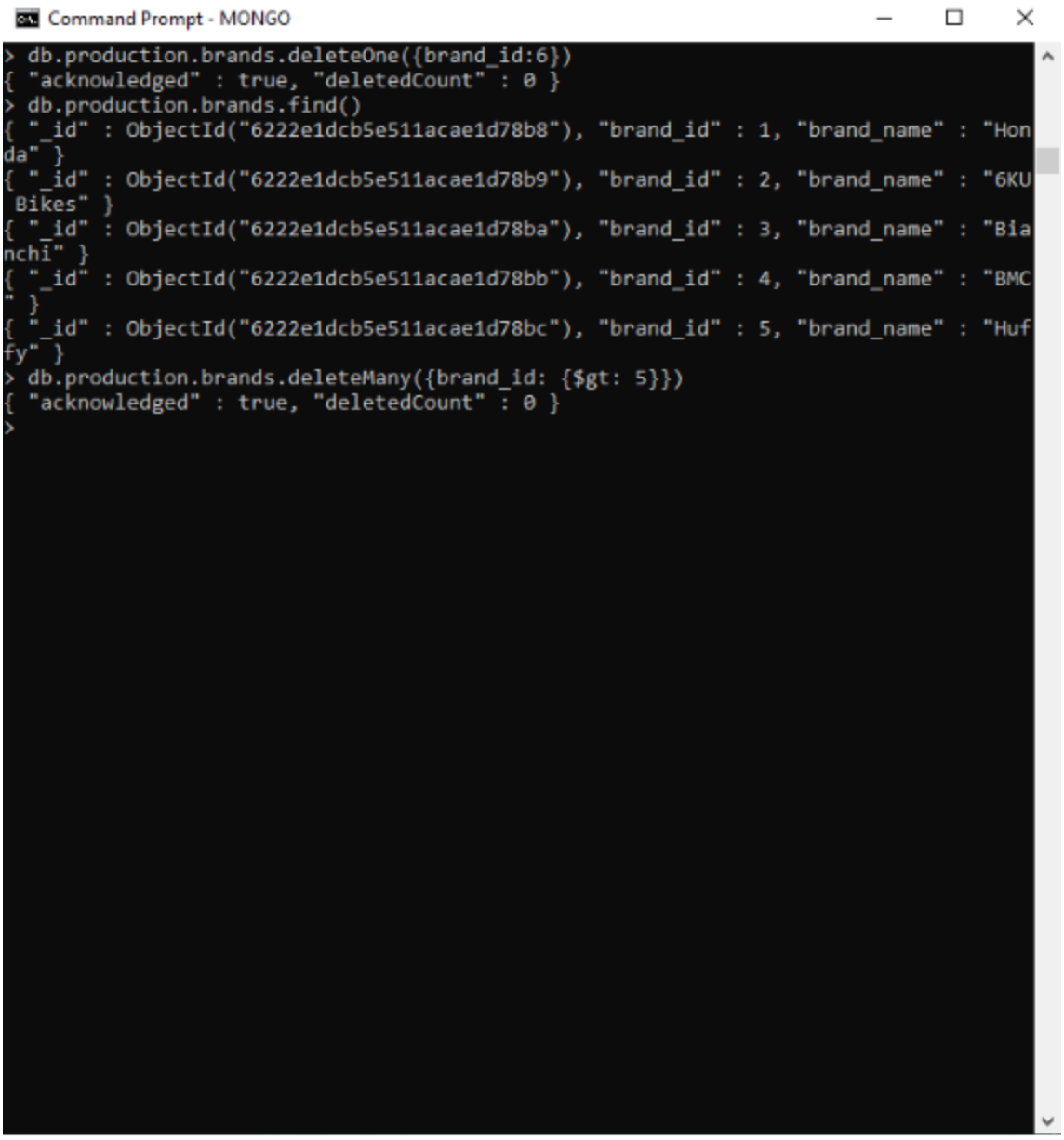


2.deleteMany()

Example –

db.production.brands.find() - to show current documents in collection

db.production.brands.deleteMany({brand_id: {$gt: 5}}) - delete documents for which brand id is greater than 5


Command Prompt - MONGO
> db.production.brands.deleteOne({brand_id:6})
{ "acknowledged" : true, "deletedCount" : 0 }
> db.production.brands.find()
{ "_id" : ObjectId("6222e1dcb5e511acae1d78b8"), "brand_id" : 1, "brand_name" : "Honda" }
{ "_id" : ObjectId("6222e1dcb5e511acae1d78b9"), "brand_id" : 2, "brand_name" : "6KU Bikes" }
{ "_id" : ObjectId("6222e1dcb5e511acae1d78ba"), "brand_id" : 3, "brand_name" : "Bianchi" }
{ "_id" : ObjectId("6222e1dcb5e511acae1d78bb"), "brand_id" : 4, "brand_name" : "BMC" }
{ "_id" : ObjectId("6222e1dcb5e511acae1d78bc"), "brand_id" : 5, "brand_name" : "Huffy" }
> db.production.brands.deleteMany({brand_id: {$gt: 5}})
{ "acknowledged" : true, "deletedCount" : 0 }
>