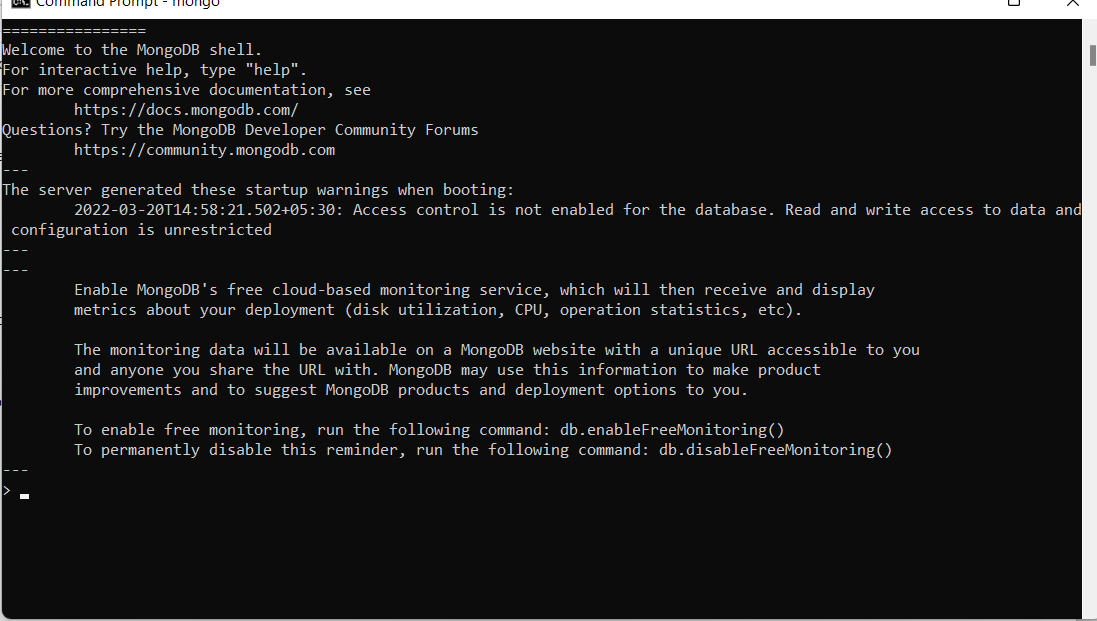
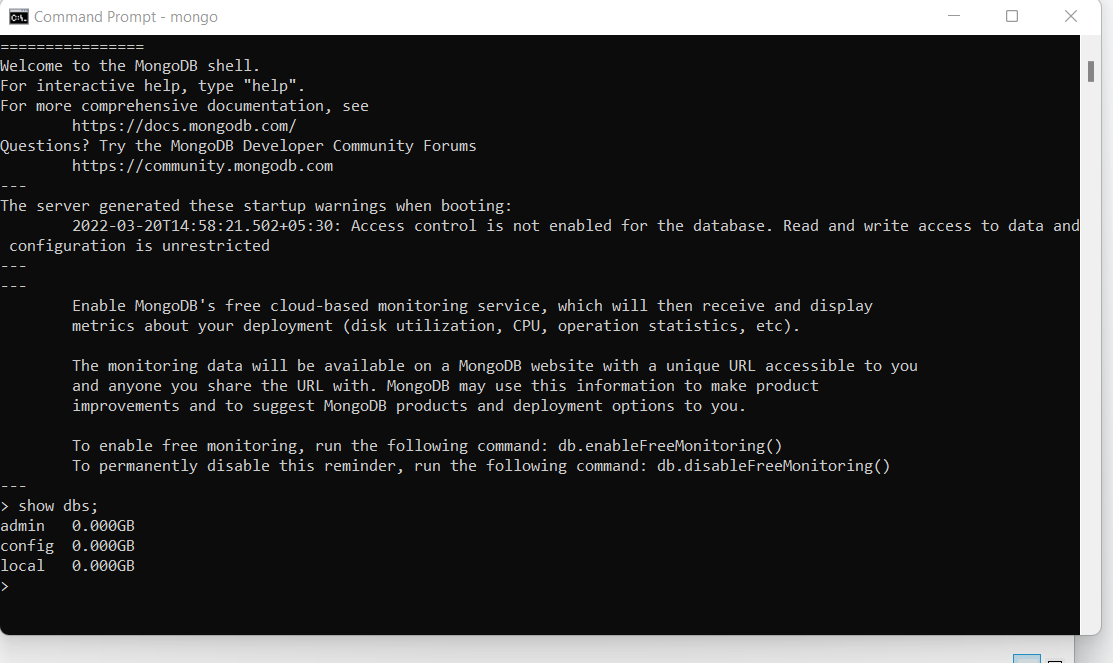
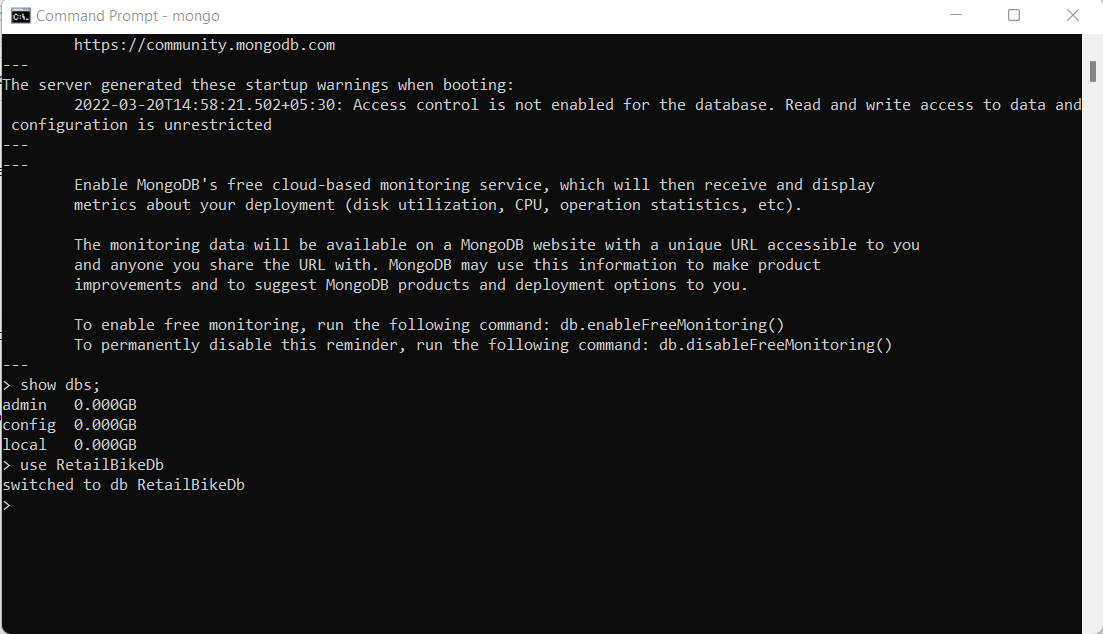
1)First start the mongo Db using command **mongo**

****

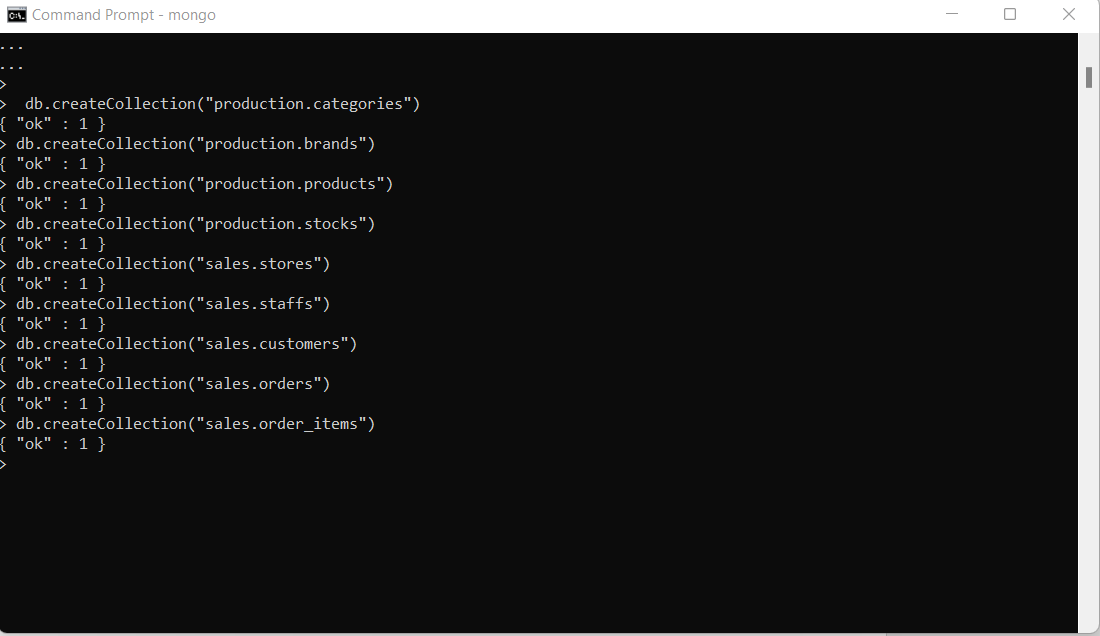
show dbs;

****

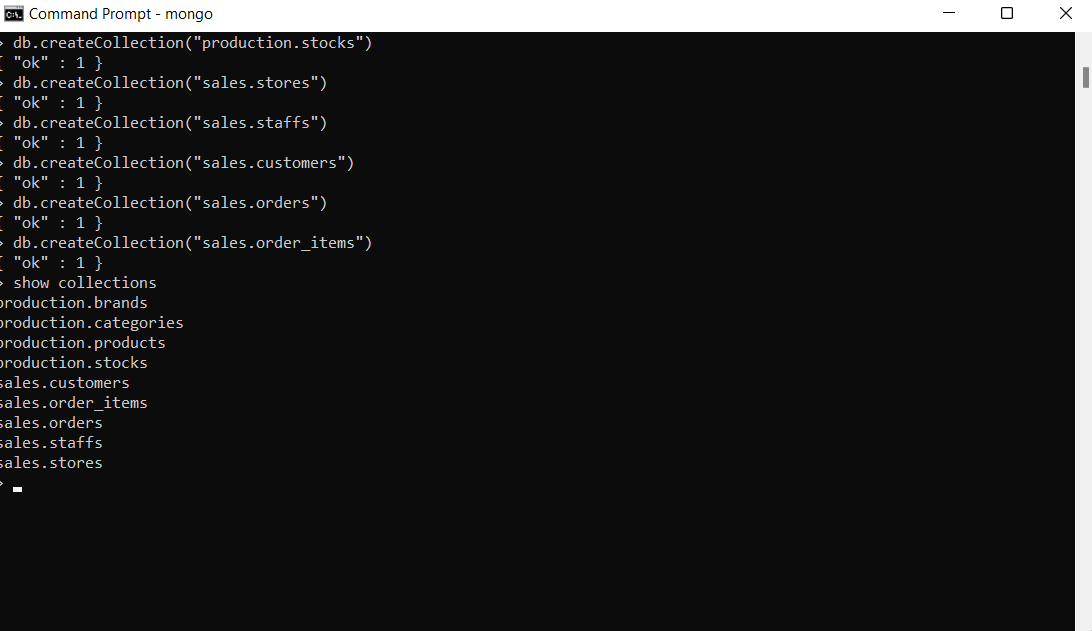
2) use RetailBikeDb

****

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"

},

{

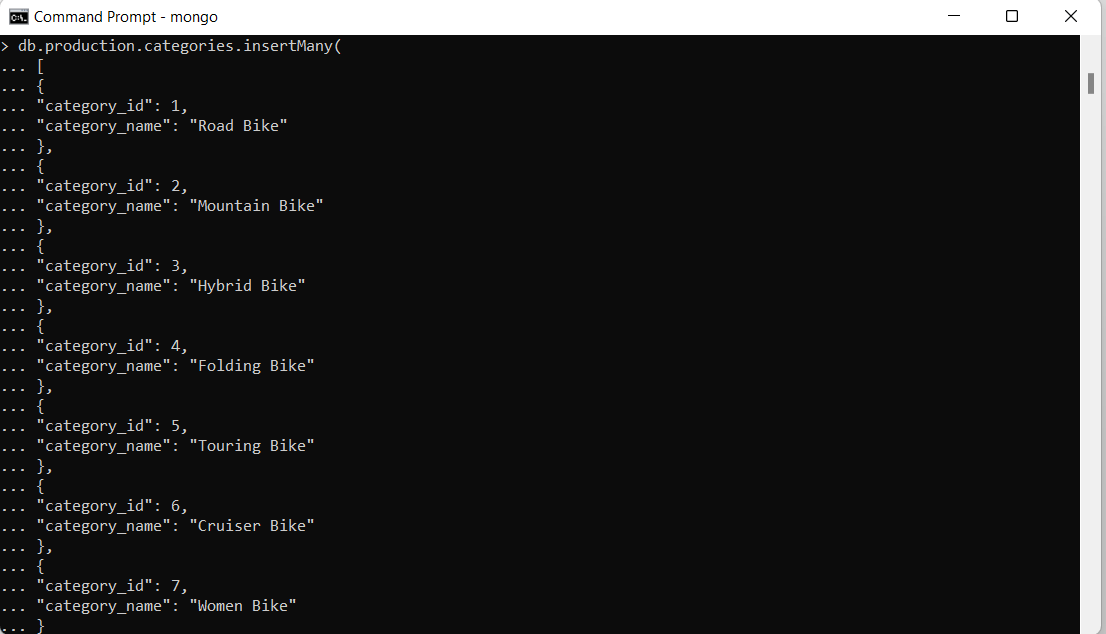
"category\_id": 7,

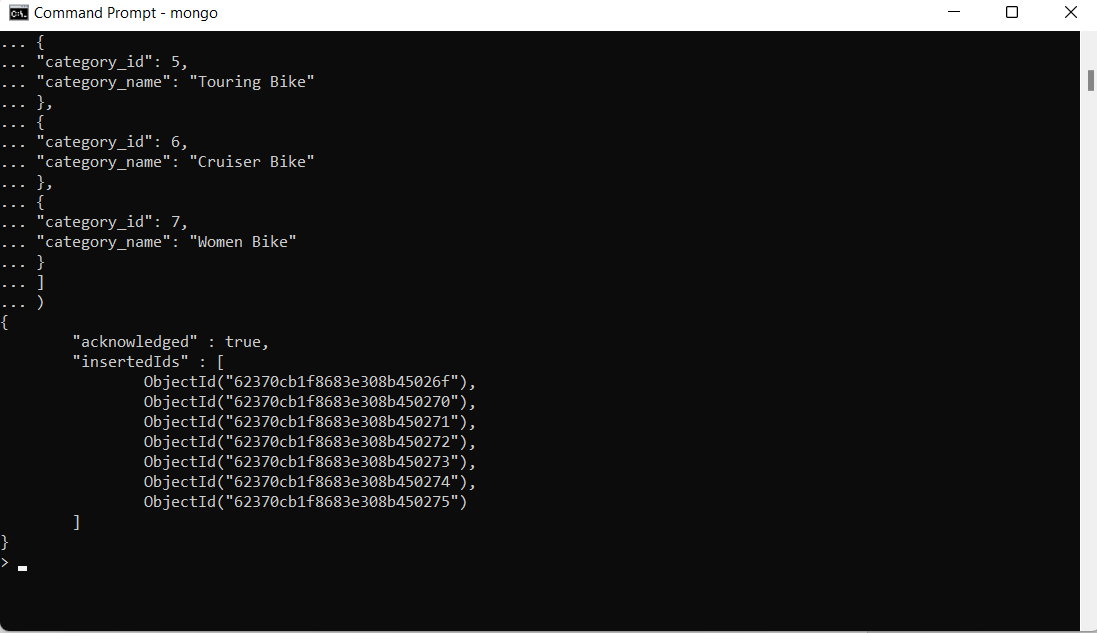
"category\_name": "Women Bike"

}

]

)





1. 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,

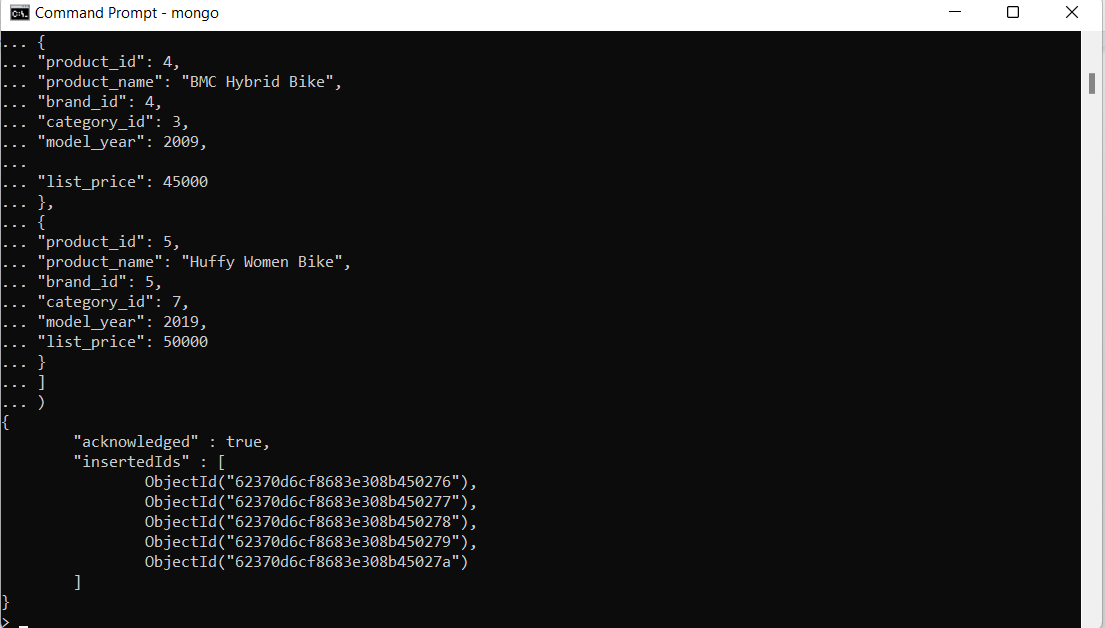
"model\_year": 2019,

"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",

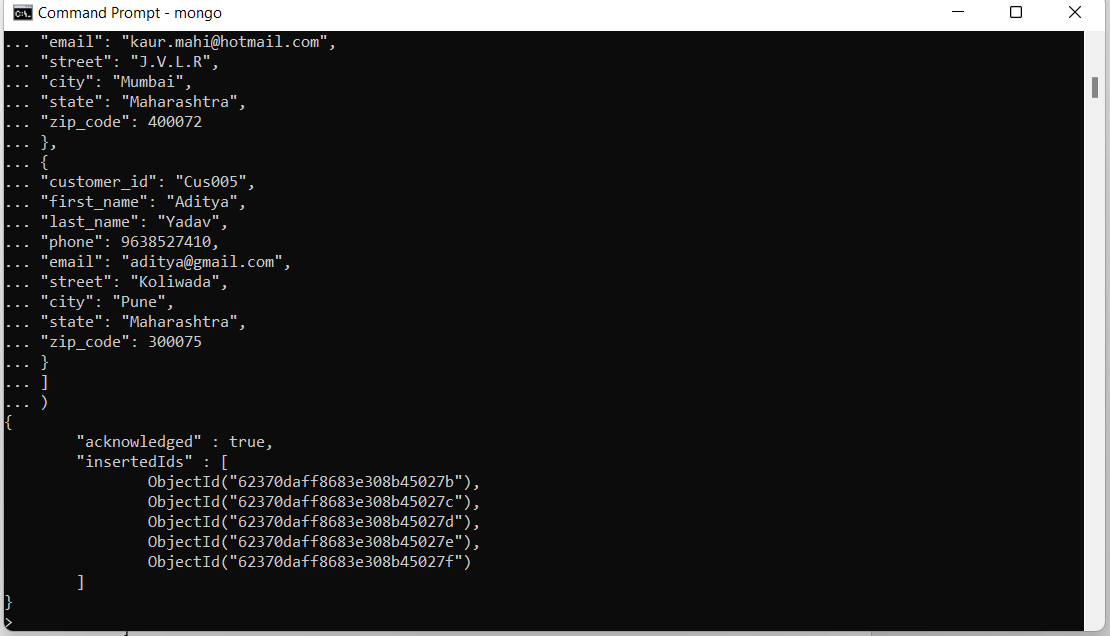
"state": "Maharashtra",

"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,

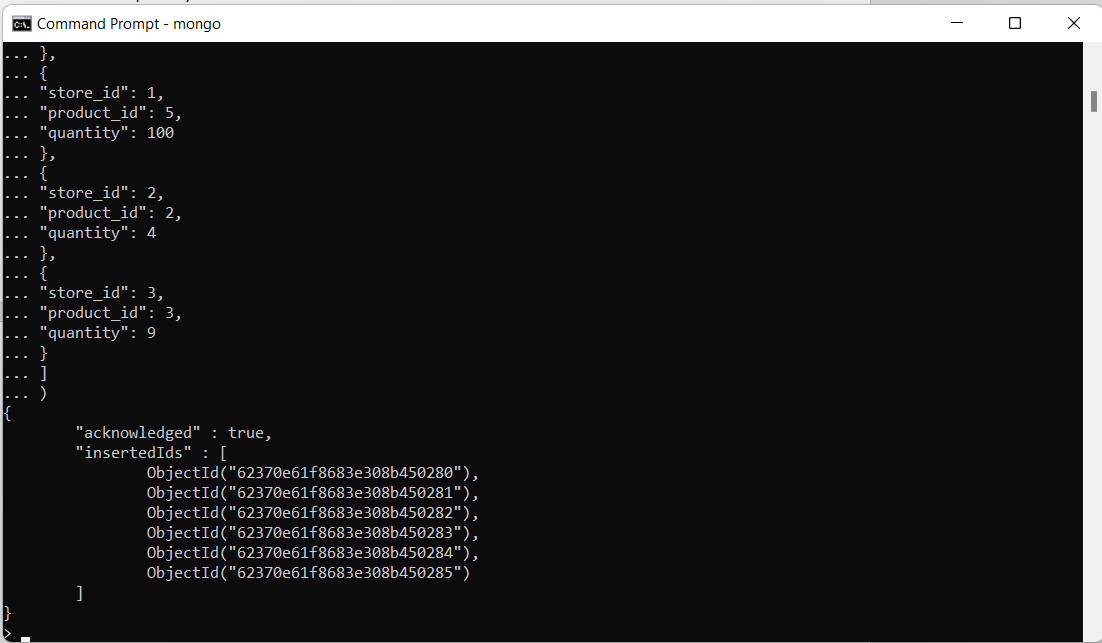
"product\_id": 3,

"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,

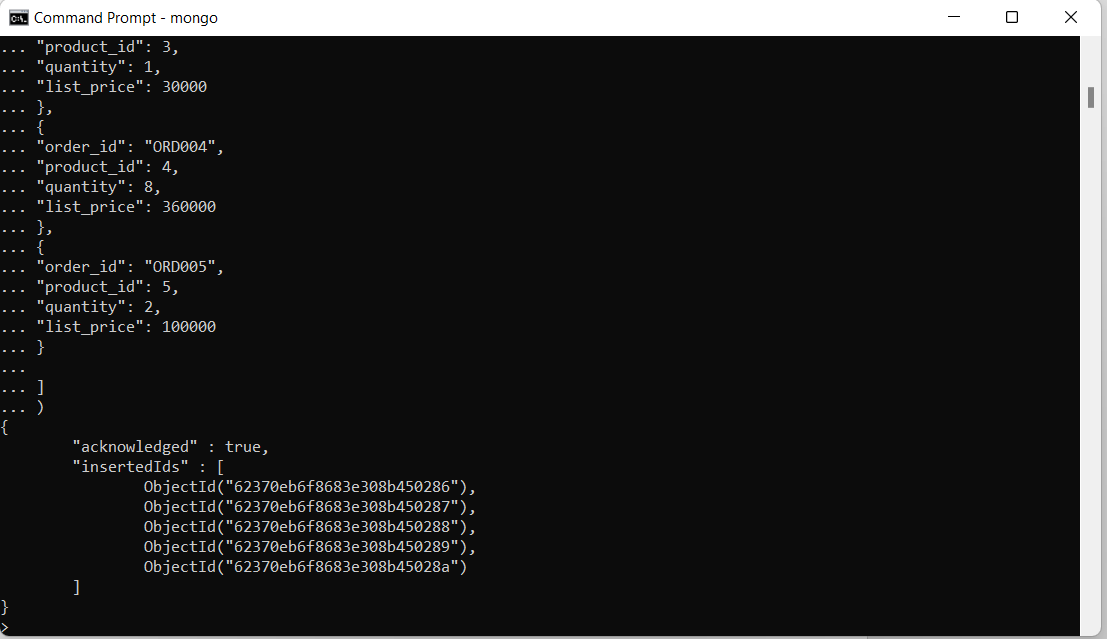
"quantity": 2,

"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,

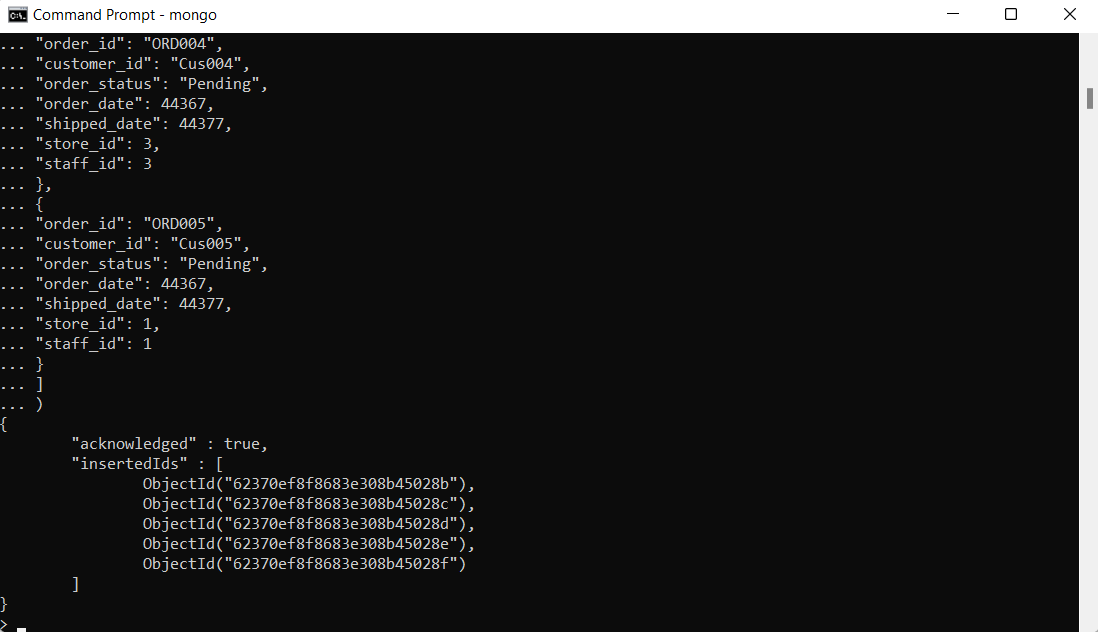
"store\_id": 1,

"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",

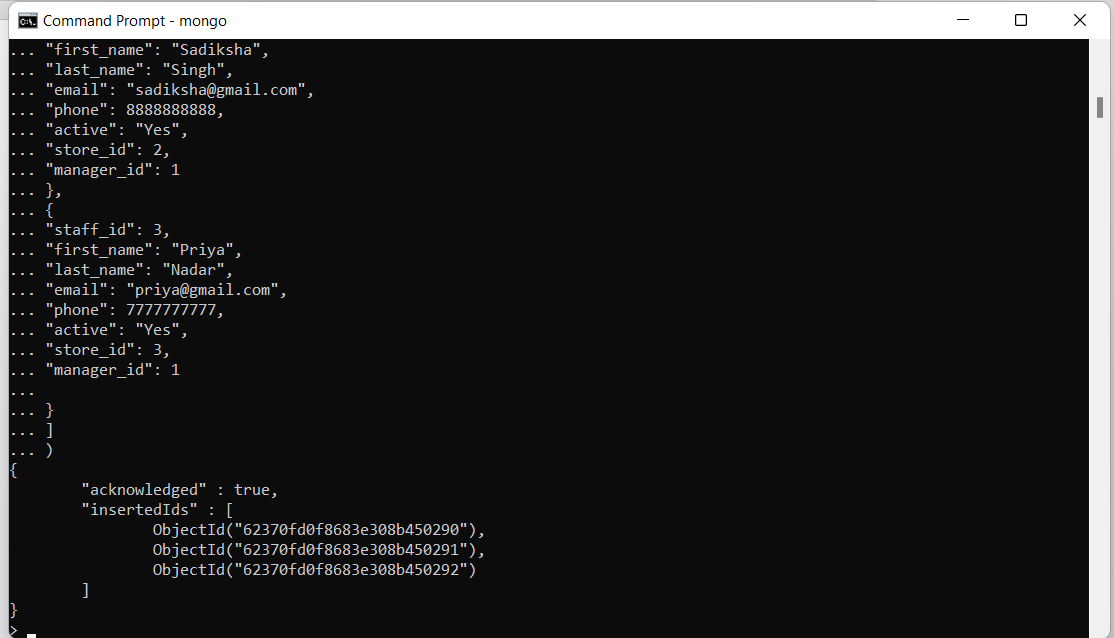
"store\_id": 3,

"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"

},

{

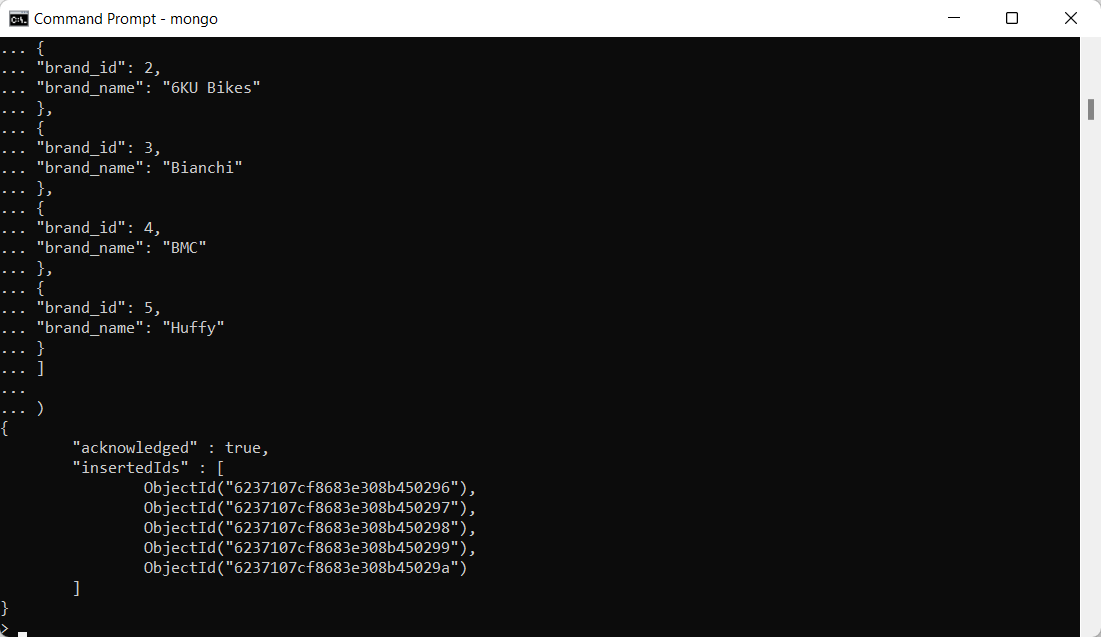
"brand\_id": 5,

"brand\_name": "Huffy"

}

]

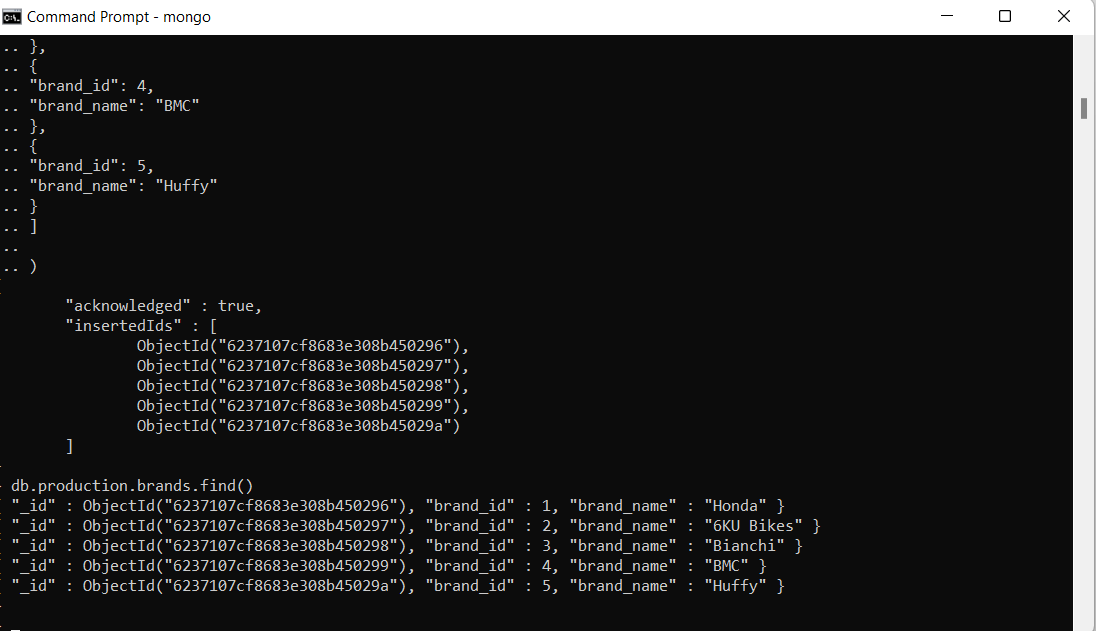
)



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**

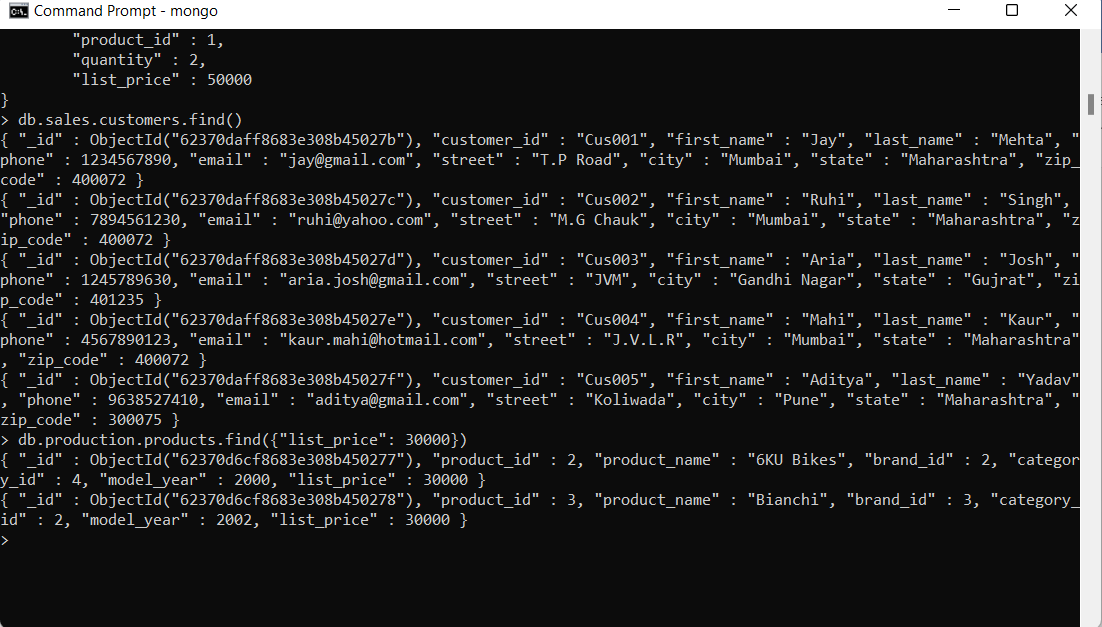
1. **Specify Equality Condition**

**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





1. **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"

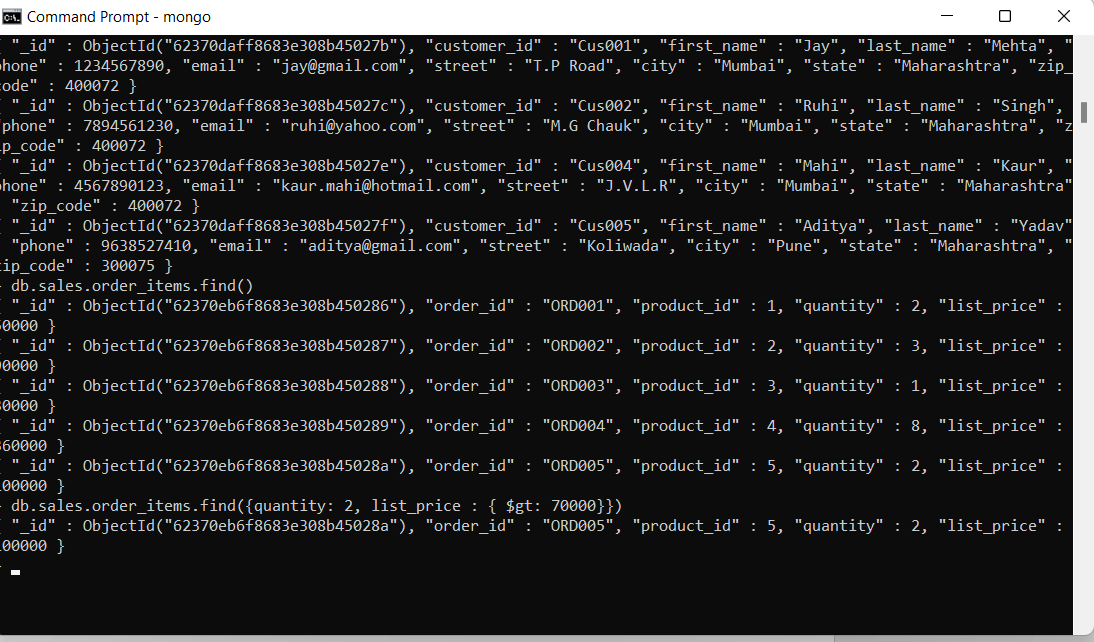


**C . Specify AND Conditions**

**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



* 1. **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



1. Update Operations

For MongoDB CRUD, there are three different methods of updating documents:

* db.collection.updateOne()
* db.collection.updateMany()
* db.collection.replaceOne()

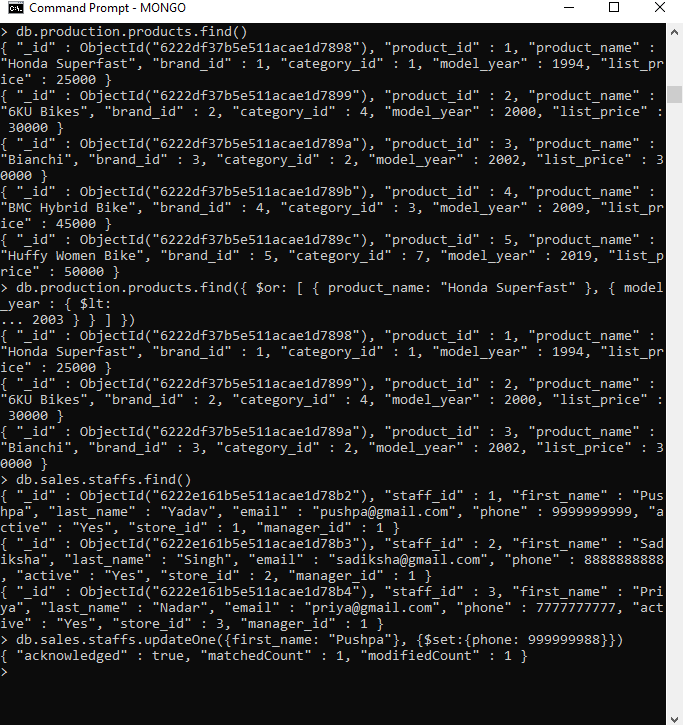
1. updateOne()

**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



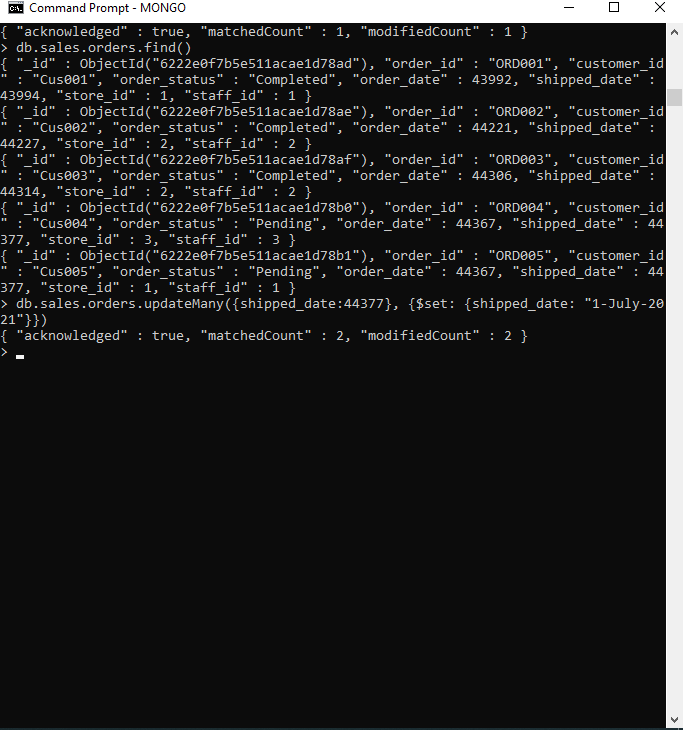
1. updateMany()

**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()

1. Delete Operations

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

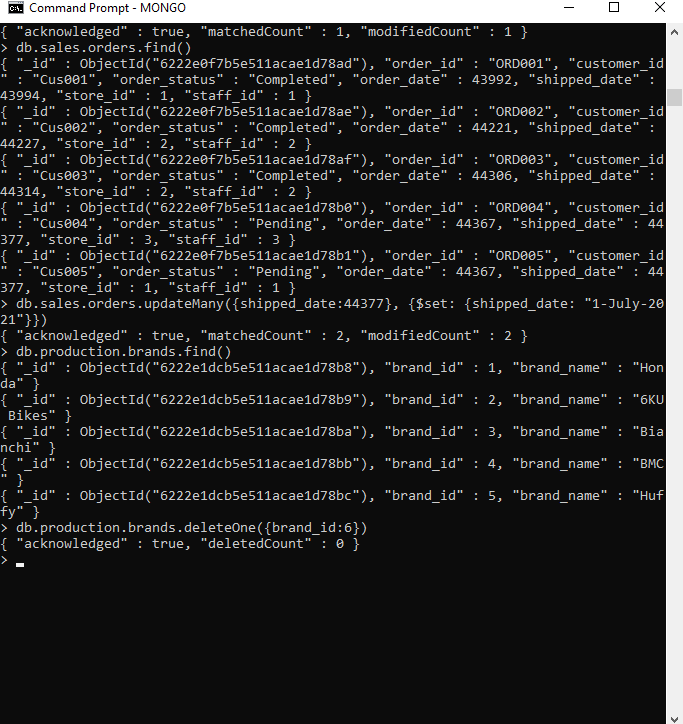
* db.collection.delete004Fne()
* db.collection.deleteMany()

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

