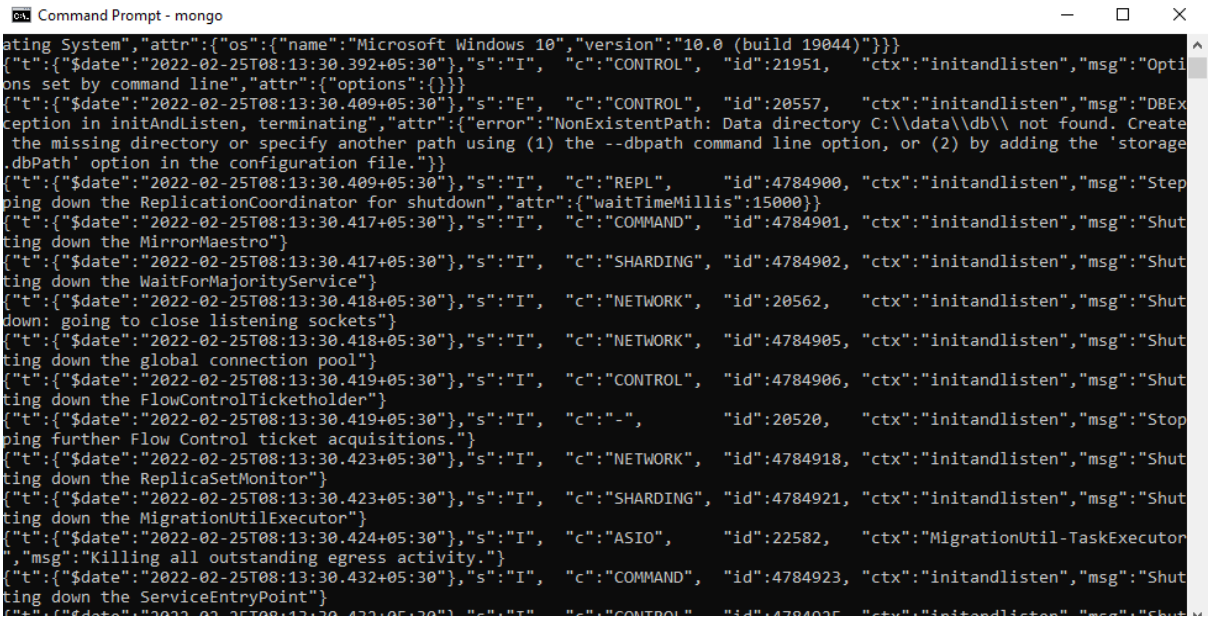
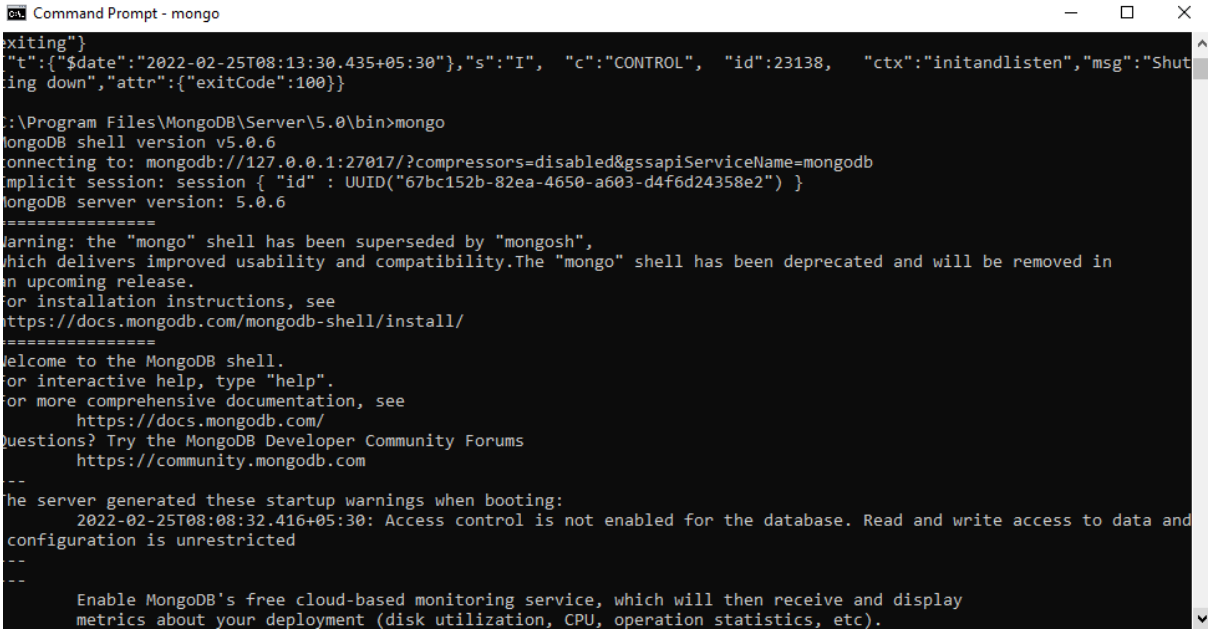
Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

Subject: Big Data Technology
Practical_6: Handling Unstructured

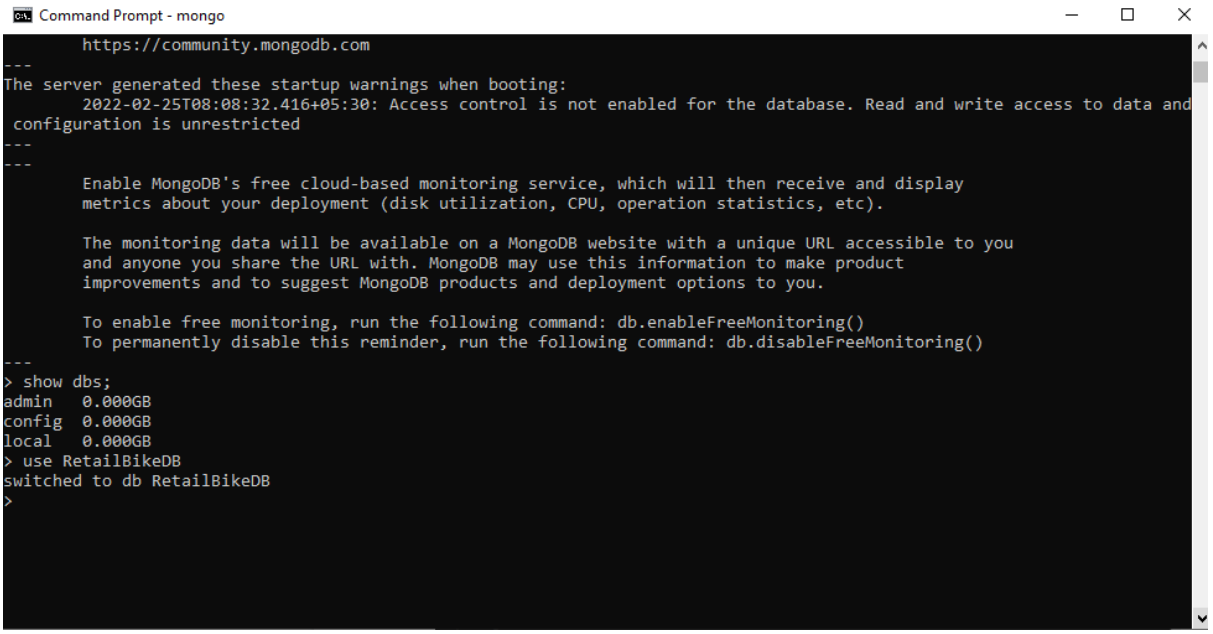
Open the command prompt and run the cd command to change directory and run the command "mongod" to start the server and run the command "mongo" to work mongoDB.



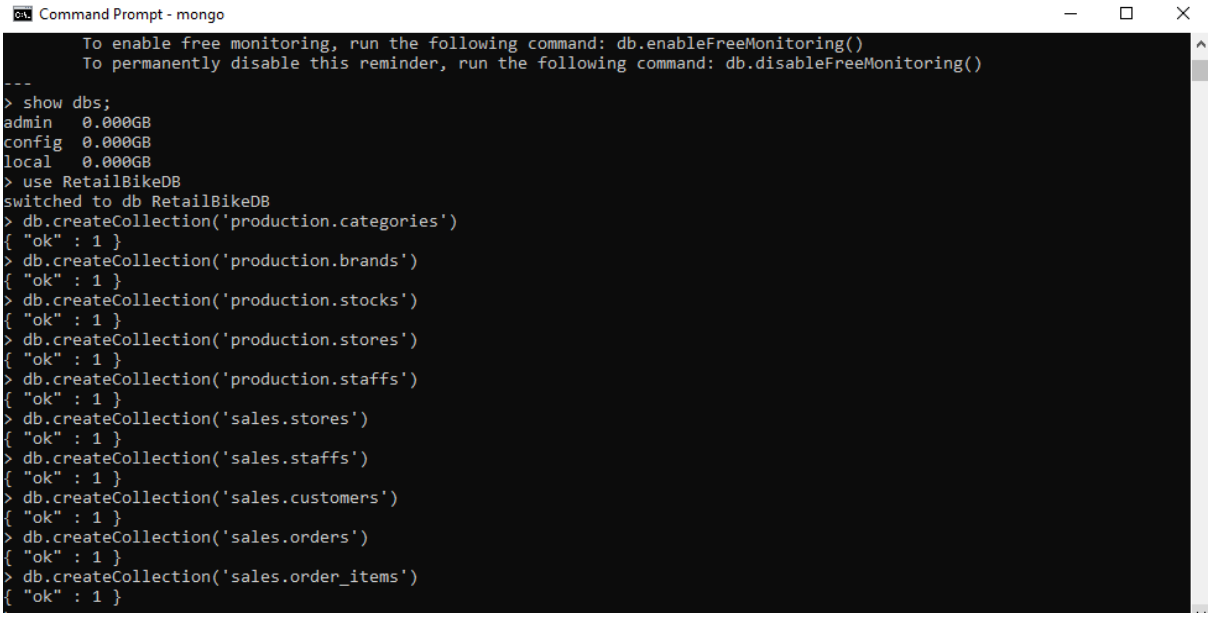


Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

Show list of databases



Creation of collection can be done using db.createCollection(name)



Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

We can show list of collection using "Show collections" commands in MongoDB.

```
> show collections
production.brands
production.categories
production.staffs
production.stocks
production.stores
sales.customers
sales.order_items
sales.orders
sales.staffs
sales.stores
>
```

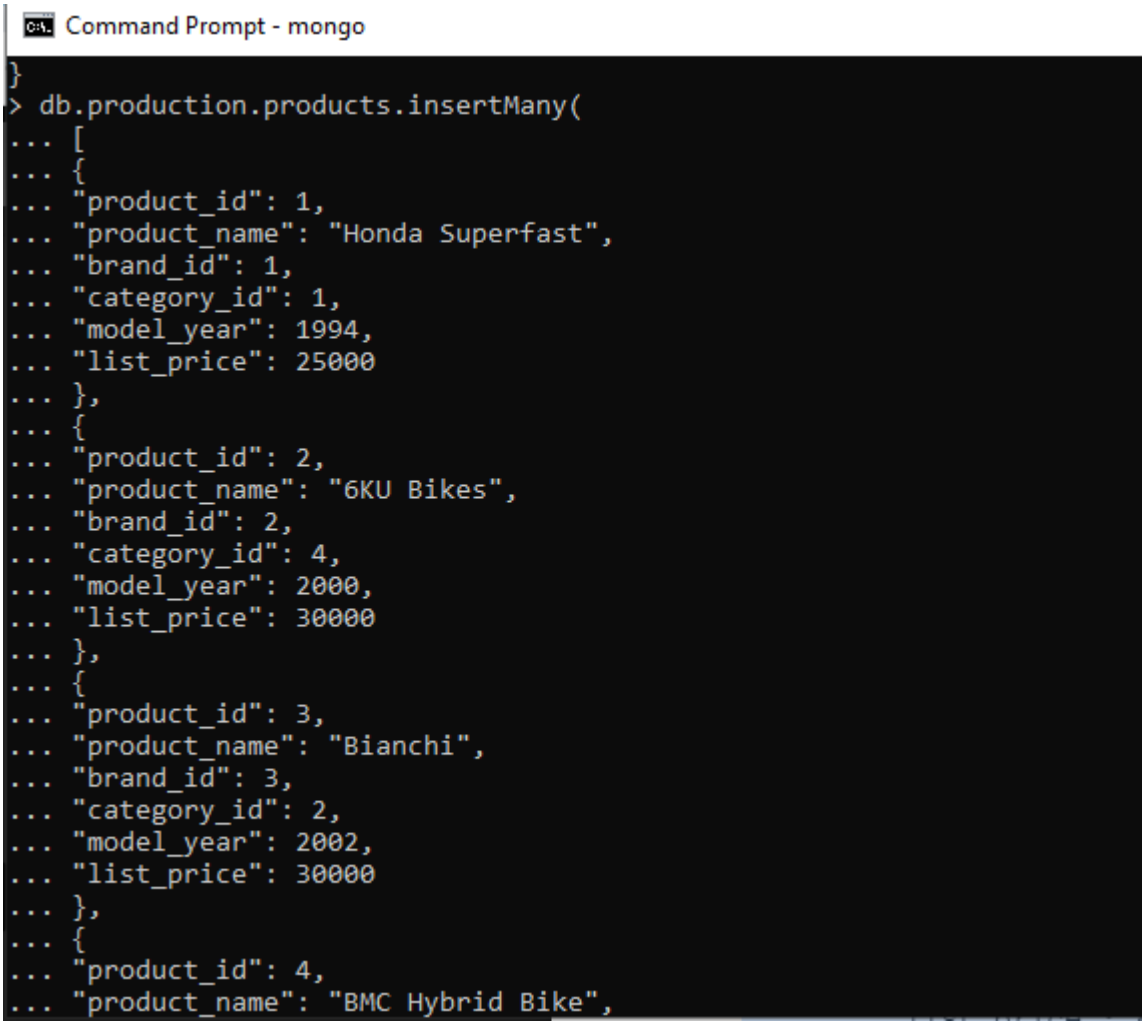
1. production.categories

Select Command Prompt - mongo

```
> 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"
... }
... ]
... )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62184a406080c37f46a3ad74"),
                ObjectId("62184a406080c37f46a3ad75"),
                ObjectId("62184a406080c37f46a3ad76"),
                ObjectId("62184a406080c37f46a3ad77"),
                ObjectId("62184a406080c37f46a3ad78"),
                ObjectId("62184a406080c37f46a3ad79"),
                ObjectId("62184a406080c37f46a3ad7a")
        ]
}
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

2. production.products



Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

Command Prompt - mongo

```
... {
... "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
... }
... ]
... )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62184ab76080c37f46a3ad7b"),
                ObjectId("62184ab76080c37f46a3ad7c"),
                ObjectId("62184ab76080c37f46a3ad7d"),
                ObjectId("62184ab76080c37f46a3ad7e"),
                ObjectId("62184ab76080c37f46a3ad7f")
        ]
}
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

3. production.stocks

Command Prompt - mongo

```
> 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,
```

Command Prompt - mongo

```
... {
... "store_id": 3,
... "product_id": 3,
... "quantity": 9
... }
... ]
... )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62184b446080c37f46a3ad80"),
                ObjectId("62184b446080c37f46a3ad81"),
                ObjectId("62184b446080c37f46a3ad82"),
                ObjectId("62184b446080c37f46a3ad83"),
                ObjectId("62184b446080c37f46a3ad84"),
                ObjectId("62184b446080c37f46a3ad85")
        ]
}
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

4. sales.customers

Command Prompt - mongo

```
}
> 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,
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

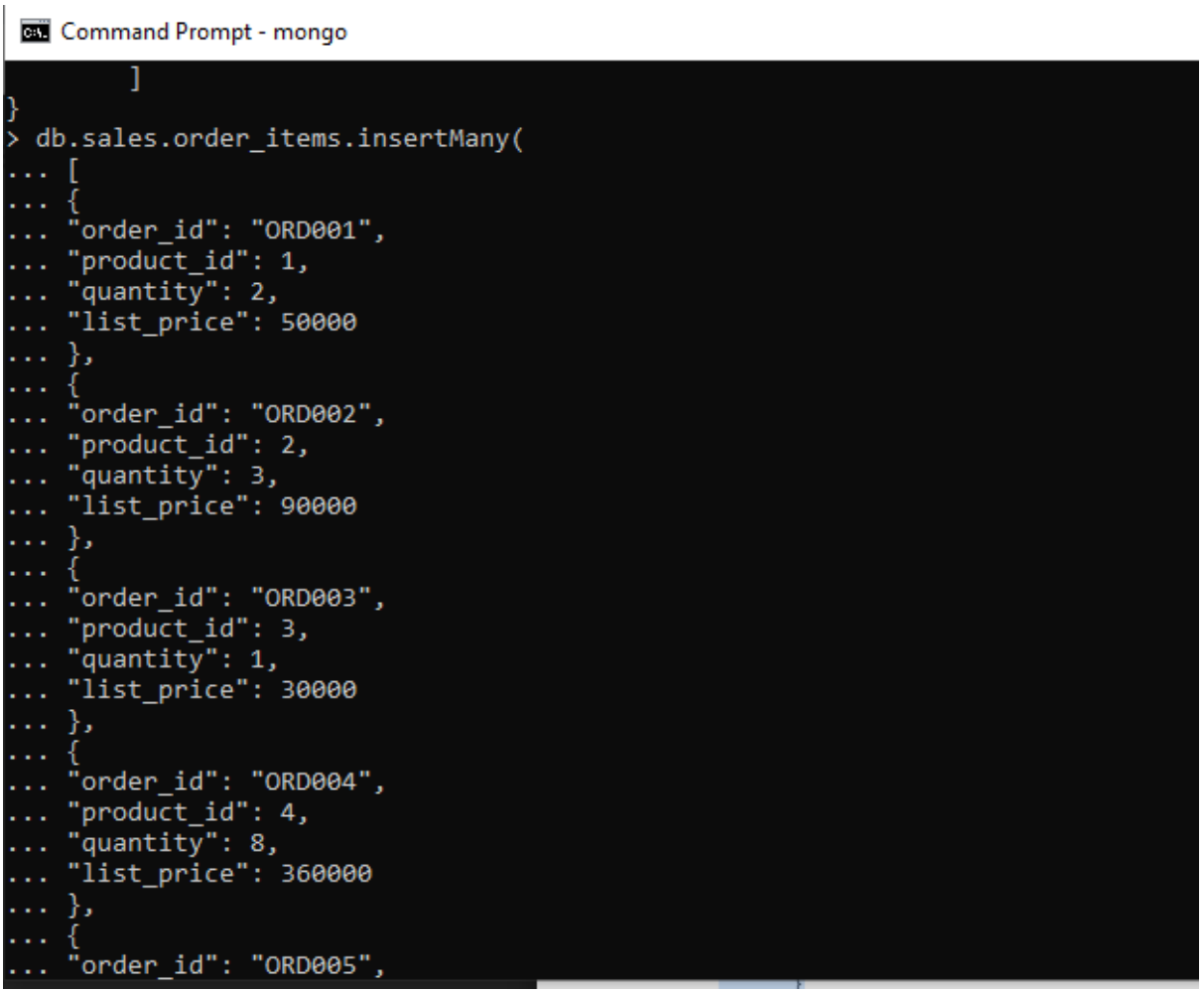
Command Prompt - mongo

```
... "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
... }
... ]
```

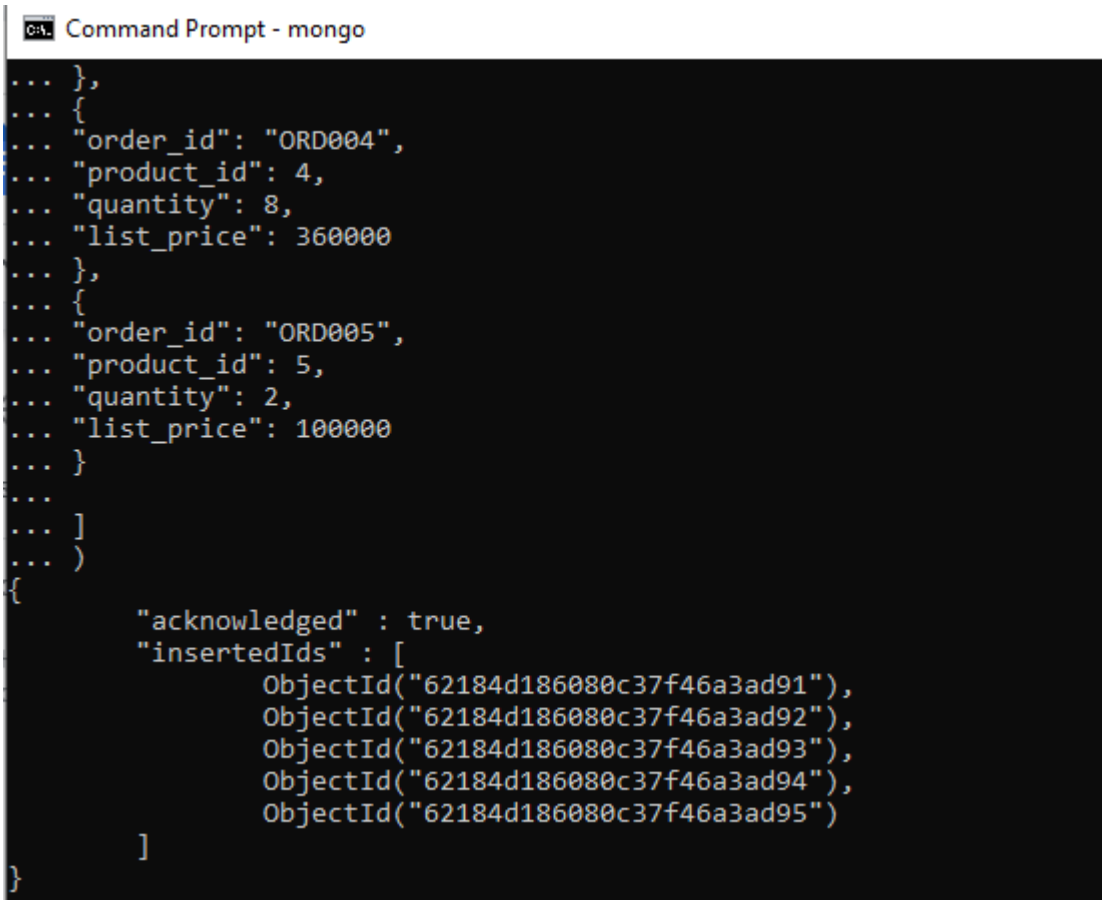
```
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62184c366080c37f46a3ad8c"),
                ObjectId("62184c366080c37f46a3ad8d"),
                ObjectId("62184c366080c37f46a3ad8e"),
                ObjectId("62184c366080c37f46a3ad8f"),
                ObjectId("62184c366080c37f46a3ad90")
        ]
}
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

5. sales.order_items

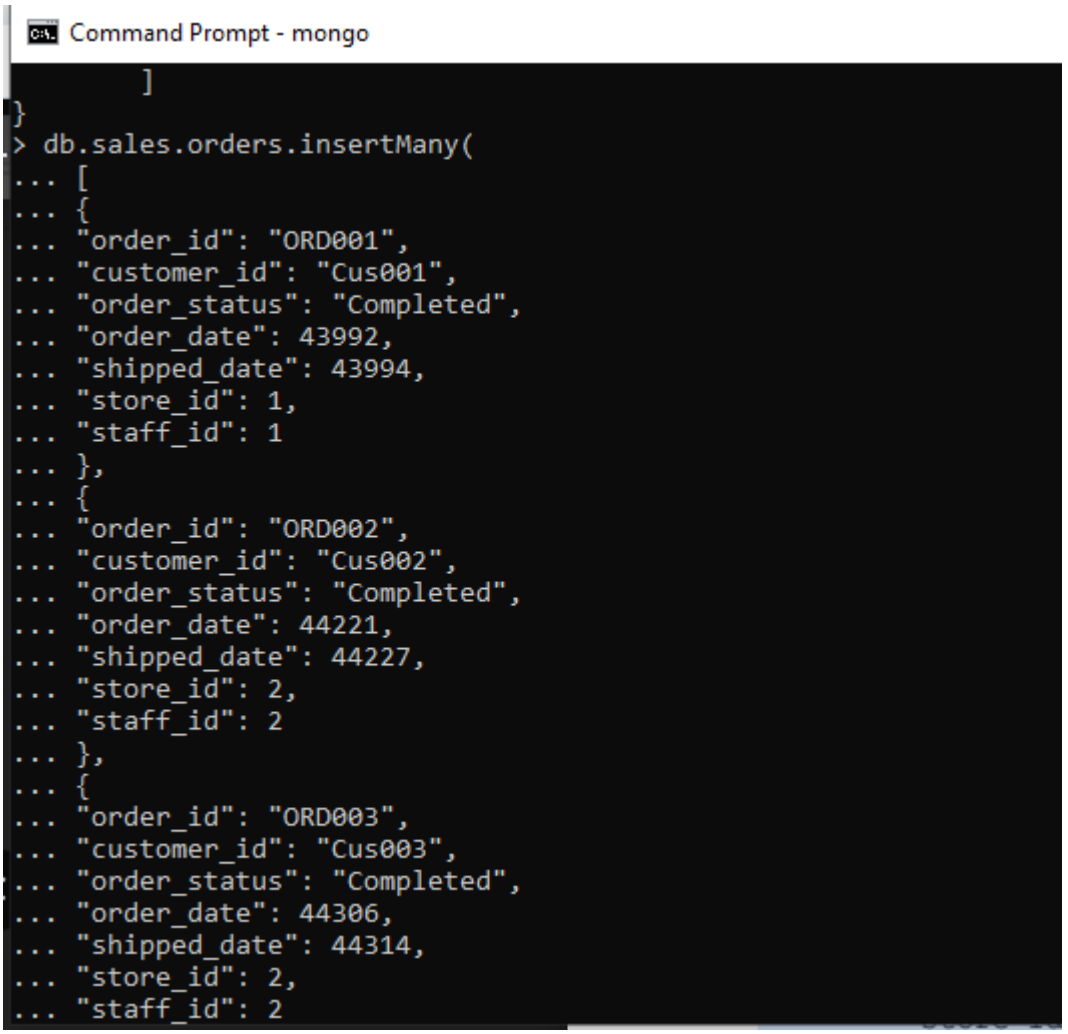


Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2



Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

6. Sales.orders



Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

Command Prompt - mongo

```
... "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
... }
... ]
... )
```

```
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62184d6a6080c37f46a3ad96"),
                ObjectId("62184d6a6080c37f46a3ad97"),
                ObjectId("62184d6a6080c37f46a3ad98"),
                ObjectId("62184d6a6080c37f46a3ad99"),
                ObjectId("62184d6a6080c37f46a3ad9a")
        ]
}
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

7. sales.staffs

Command Prompt - mongo

```
}
> 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",
... "phone": 7777777777,
... "active": "Yes",
... "store_id": 3,
... "manager_id": 1
...
... }
... ]
... )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62184e3f6080c37f46a3ad9b"),
                ObjectId("62184e3f6080c37f46a3ad9c"),
                ObjectId("62184e3f6080c37f46a3ad9d")
        ]
}
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

8. sales.stores

Command Prompt - mongo

```
}
> 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",
...
... "street": "M.G",
... "city": "Gandhi Nagar",
... "state": "Gujrat",
... "zip_code": 401235
... }
... ]
... )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62184e886080c37f46a3ad9e"),
                ObjectId("62184e886080c37f46a3ad9f"),
                ObjectId("62184e886080c37f46a3ada0")
        ]
}
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

9. Production.brands

Command Prompt - mongo

```
> 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"
... }
... ]
...
... )
```

```
... )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("62184ecc6080c37f46a3ada1"),
                ObjectId("62184ecc6080c37f46a3ada2"),
                ObjectId("62184ecc6080c37f46a3ada3"),
                ObjectId("62184ecc6080c37f46a3ada4"),
                ObjectId("62184ecc6080c37f46a3ada5")
        ]
}
```

If you want to get more specific with a read operation and find a desired subsection of the records, you can use the previously mentioned filtering criteria to choose what results should be returned. One of the most common ways of filtering the results is to search by value.
Example - db.production.brands.find({"brand_name" : "Honda"})

Command Prompt - mongo

```
}
> show dbs;
RetailBikeDB  0.000GB
admin         0.000GB
config        0.000GB
local         0.000GB
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

findOne()

In order to get one document that satisfies the search criteria, we can simply use the findOne() method on our chosen collection. If multiple documents satisfy the query, this method returns the first document according to the natural order which reflects the order of documents on the disk. If no documents satisfy the search criteria, the function returns null. The function takes the following form of syntax.

Syntax- db.{collection}.findOne({query}, {projection})

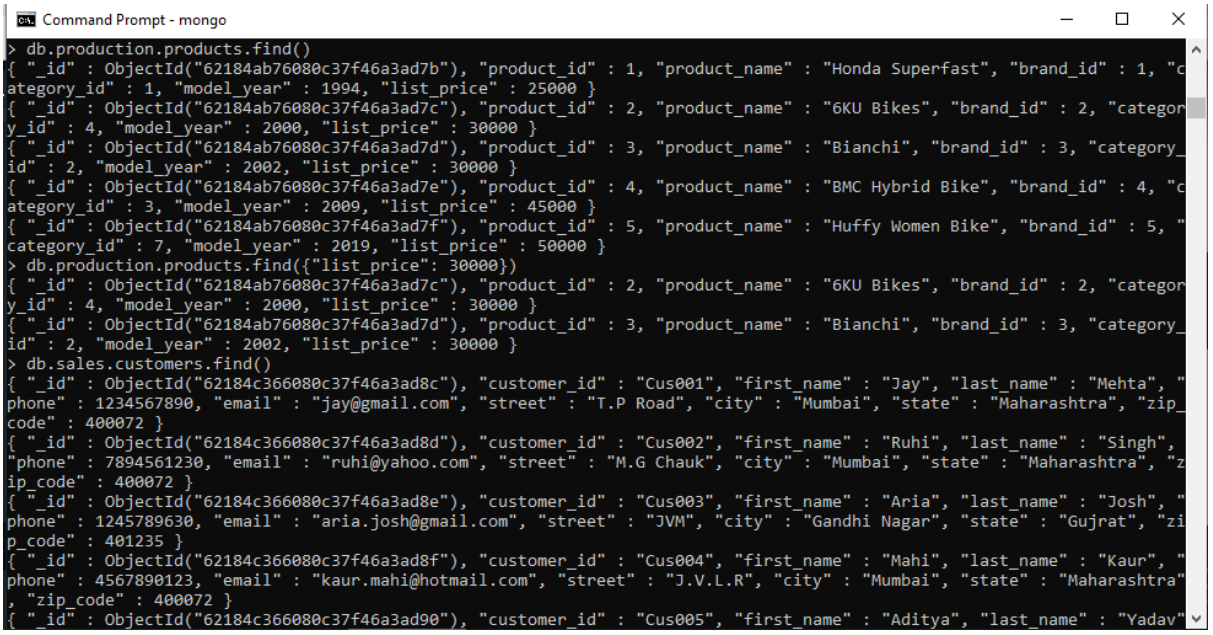
Example – db.sales.order_items.findOne({"quantity": 2})

```
> db.production.brands.find({'brand_name':'Honda'})
{ "_id" : ObjectId("62184ecc6080c37f46a3ada1"), "brand_id" : 1, "brand_name" : "Honda" }
> db.sales.order_items.findOne({"quantity": 2})
{
        "_id" : ObjectId("62184d186080c37f46a3ad91"),
        "order_id" : "ORD001",
        "product_id" : 1,
        "quantity" : 2,
        "list_price" : 50000
}
```

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"

```
> db.sales.customers.find({city: { $in: [ "Mumbai", "Pune" ] } })
{ "_id" : ObjectId("62184c366080c37f46a3ad8c"), "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 }
{ "_id" : ObjectId("62184c366080c37f46a3ad8d"), "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 }
{ "_id" : ObjectId("62184c366080c37f46a3ad8f"), "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 }
{ "_id" : ObjectId("62184c366080c37f46a3ad90"), "customer_id" : "Cus005", "first_name" : "Aditya", "last_name" : "Yadav", "phone" : 9638527410, "email" : "aditya@gmail.com", "street" : "Koliwada", "city" : "Pune", "state" : "Maharashtra", "zip_code" : 300075 }
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

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

```
> db.sales.order_items.find()
{ "_id" : ObjectId("62184d186080c37f46a3ad91"), "order_id" : "ORD001", "product_id" : 1, "quantity" : 2, "list_price" :
50000 }
{ "_id" : ObjectId("62184d186080c37f46a3ad92"), "order_id" : "ORD002", "product_id" : 2, "quantity" : 3, "list_price" :
90000 }
{ "_id" : ObjectId("62184d186080c37f46a3ad93"), "order_id" : "ORD003", "product_id" : 3, "quantity" : 1, "list_price" :
30000 }
{ "_id" : ObjectId("62184d186080c37f46a3ad94"), "order_id" : "ORD004", "product_id" : 4, "quantity" : 8, "list_price" :
360000 }
{ "_id" : ObjectId("62184d186080c37f46a3ad95"), "order_id" : "ORD005", "product_id" : 5, "quantity" : 2, "list_price" :
100000 }
```

```
> db.sales.order_items.find({quantity: 2, list_price : { $gt: 70000}})
{ "_id" : ObjectId("62184d186080c37f46a3ad95"), "order_id" : "ORD005", "product_id" : 5, "quantity" : 2, "list_price" :
100000 }
```

d)Specify OR Conditions

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

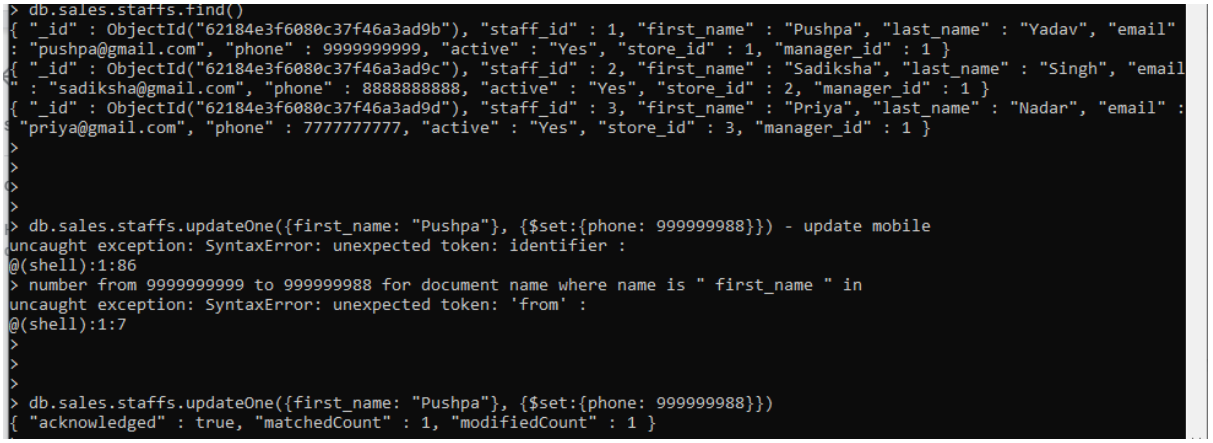
> db.production.products.find({ $or: [{ product_name: "Honda Superfast" }, { model_year : { $lt: 2003 } }] }) - to show all the document which is either "product name" as "Honda Superfast" or "model year" is less than year 2003

```
> db.production.products.find()
{ "_id" : ObjectId("62184ab76080c37f46a3ad7b"), "product_id" : 1, "product_name" : "Honda Superfast", "brand_id" : 1, "c
ategory_id" : 1, "model_year" : 1994, "list_price" : 25000 }
{ "_id" : ObjectId("62184ab76080c37f46a3ad7c"), "product_id" : 2, "product_name" : "6KU Bikes", "brand_id" : 2, "categor
y_id" : 4, "model_year" : 2000, "list_price" : 30000 }
{ "_id" : ObjectId("62184ab76080c37f46a3ad7d"), "product_id" : 3, "product_name" : "Bianchi", "brand_id" : 3, "category_
id" : 2, "model_year" : 2002, "list_price" : 30000 }
{ "_id" : ObjectId("62184ab76080c37f46a3ad7e"), "product_id" : 4, "product_name" : "BMC Hybrid Bike", "brand_id" : 4, "c
ategory_id" : 3, "model_year" : 2009, "list_price" : 45000 }
{ "_id" : ObjectId("62184ab76080c37f46a3ad7f"), "product_id" : 5, "product_name" : "Huffy Women Bike", "brand_id" : 5, "
category_id" : 7, "model_year" : 2019, "list_price" : 50000 }
```

```
> db.production.products.find({ $or: [ { product_name: "Honda Superfast" }, { model_year : { $lt:
... 2003 } } ] })
{ "_id" : ObjectId("62184ab76080c37f46a3ad7b"), "product_id" : 1, "product_name" : "Honda Superfast", "brand_id" : 1, "c
ategory_id" : 1, "model_year" : 1994, "list_price" : 25000 }
{ "_id" : ObjectId("62184ab76080c37f46a3ad7c"), "product_id" : 2, "product_name" : "6KU Bikes", "brand_id" : 2, "categor
y_id" : 4, "model_year" : 2000, "list_price" : 30000 }
{ "_id" : ObjectId("62184ab76080c37f46a3ad7d"), "product_id" : 3, "product_name" : "Bianchi", "brand_id" : 3, "category_
id" : 2, "model_year" : 2002, "list_price" : 30000 }
```

Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

updateOne()
We can update a currently existing record and change a single document with an update operation. To do this, we use the updateOne() method on a chosen collection. To update a document, we provide the method with two arguments: an update filter and an update action.
The update filter defines which items we want to update, and the update action defines how to update those items. We first pass in the update filter. Then, we use the "$set" key and provide the fields we want to update as a value. This method will update the first record that matches the provided filter.
Example –
> db.sales.staffs.find() - to show currect document in the system
> db.sales.staffs.updateOne({first_name: "Pushpa"}, {$set:{phone: 999999988}}) - update mobile number from 9999999999 to 999999988 for document name where name is " first_name " in collection



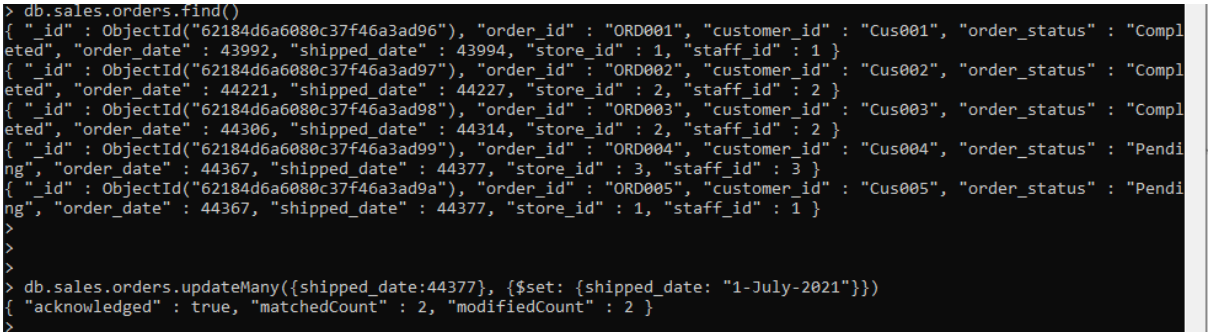
Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

updateMany() allows us to update multiple items by passing in a list of items, just as we did when inserting multiple items. This update operation uses the same syntax for updating a single document.

Example –

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



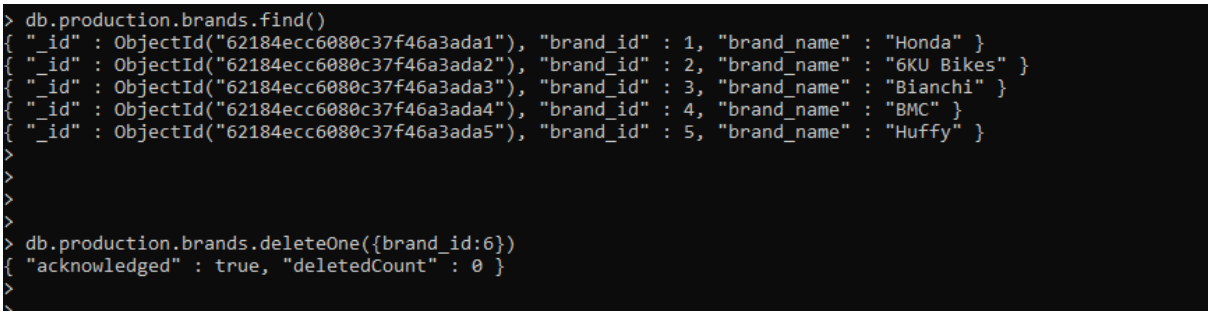
deleteOne()

deleteOne() is used to remove a document from a specified collection on the MongoDB server. A filter criteria is used to specify the item to delete. It deletes the first record that matches the provided filter.

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



Name: Pooja Pathak
Roll No: 14
MSc Part-I Sem-2

deleteMany()
deleteMany() is a method used to delete multiple documents from a desired collection with a single delete operation. A list is passed into the method and the individual items are defined with filter criteria as in deleteOne().
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