```
# There is no “create” command in the MongoDB Shell.
# In order to create a database, you will first need to switch the context to a
# non-existing database using the use command:

> show databases
#Or
> show dbs

# switch to Bookstore
> use Bookstore

# Create a collection - Books
# However, In MongoDB, create a collection is not mandatory.
# MongoDB creates collection automatically, when you insert some document.
# Here we use createCollection explicitely..!
> db.createCollection("Books");

# Verify if collection is created or not
> show collections

# insert one document to Books
> db.Books.insertOne({"title": "Book3", "author": "author3", "pages": 400, "genere":
["magical", "realism"], "price": 400, "copies":2})

# insert more than one one document to Books
> db.Books.insertMany([{"title": "Book4", "author": "author4", "pages": 400, "genere":
["magical", "realism"], "price": 400, "copies":5}, {"title": "Book5", "author": "author5",
"pages": 400, "genere": ["magical", "realism"], "price": 400, "copies":5}])

# Create a new collection - "Notes" without using createCollection command.
# MongoDB creates collection automatically, when you insert some document.
> db.Notes.insertOne({"title": "Book3", "author": "author3", "pages": 400, "genere":
["magical", "realism"], "price": 400, "copies":2})

# List all collections
> show collections

# Delete Notes from database using drop()
> db.Notes.drop()

# verify drop is done for "Notes"
> show collections

# insert one more document to Books
> db.Books.insertOne({"title": "Harry Potter 2", "author": "J K Rowling", "pages": 400,
"genere": ["magical", "realism"], "price": 400, "copies":2})

# Query1, find() method is used to Find all the documents present in the collection:.
> db.Books.find()

# Query2 - finding all document with author = "J K Rowling"
> db.Books.find({"author": "J K Rowling"})

# Query3 - finding all document with genere = "magical" and "realism" both
> db.Books.find({"genere": ["magical", "realism"]})

# Query4 - finding all document with price = 400
> db.Books.find({"price": 400})

# Query5 - finding all document with price = 400 and author = "J K Rowling"
> db.Books.find({author: "J K Rowling", price: 1000})

# Query6 - finding document with unque object id
> db.Books.find({_id: ObjectId("63bb2e9593b418ae2251d144")})
```

```
# Delete operation: Delete one document identified with unique object id
> db.Books.deleteOne({_id: ObjectId("63bb2e9593b418ae2251d144")})

# Delete operation: Delete more than one document identified with author = "J K Rowling"
> db.Books.deleteMany({"author": "J K Rowling"})

# verify documents have been deleted..!
> db.Books.find()

# Update operation: Update author to J K Rowling in the ducument with given unique object
id
> db.Books.updateOne({_id: ObjectId("63bb2e9593b418ae2251d145")}, {$set: { "author": "J K
Rowling"}})

# verify the documents with given unique object id has been updated..!
> db.Books.find({_id: ObjectId("63bb2e9593b418ae2251d145")})
```