MongoDB CRUD Report

By Arjun Raj

CRUD operations in MongoDB

➔ Create Operations

➔ Read Operations

➔ Update Operations

➔ Delete Operations

CREATE operations

➔   InsertOne()

➔   Insertmany()

➔   Insertmany() to insert more objects

Step 1 : Type 'mongo' in Command Prompt

```
Microsoft Windows [Version 10.0.18363.1237]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\goku2>mongo
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("f066e3bf-b583-4f88-8533-2df184dc9834") }
MongoDB server version: 4.4.2
---
The server generated these startup warnings when booting:
        2020-12-09T12:17:09.970+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---

        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()
---
>
```

InsertOne operation

Used to insert one element into the database.

```
---
> show dbs
admin    0.000GB
config   0.000GB
local    0.000GB
> use flightData
switched to db flightData
>
>
> db.flights.insertOne({ "departureAirport":"CHN",
... "arrivalAirport":"BGL",
... "aircraftNumber": "Boeing A330",
... "distance": 3000,
... "interContinental":false
...
... })
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5fd0e45aaacce2fb8420b43d")
}
>
```

InsertMany operation

Used to insert more than one element into the database at once.

```
>
> db.flights.insertMany([ {
... "departureAirport":"IND",
... "arrivalAirport":"LON",
... "aircraftNumber": "Airbus A380",
... "distance": 6000,
... "interContinental": true
... },
... {
...     "departureAirport":"CHN",
... "arrivalAirport":"BGL",
... "aircraftNumber": "Boeing A330",
... "distance": 3000,
... "interContinental":false
...
... },
... {
... "departureAirport":"LON",
... "arrivalAirport":"BOM",
... "aircraftNumber": "Space X Crew-Dragon",
... "distance": 6000,
... "interContinental": true
... },
... {
... "departureAirport":"IXD",
... "arrivalAirport":"JFK",
... "aircraftNumber": "Virgin Galactic",
... "distance": 6000,
... "interContinental": true
... }
... ]
... )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fd0f0ecaacce2fb8420b43e"),
                ObjectId("5fd0f0ecaacce2fb8420b43f"),
                ObjectId("5fd0f0ecaacce2fb8420b440"),
                ObjectId("5fd0f0ecaacce2fb8420b441")
        ]
}
>
```

InsertMany operation

Used to insert more than one objects into the database at once.

```
> db.flights.insertMany(
...     [
...         { name: "akshay", age: 42 },
...         { name: "boom", age: 22 },
...         { name: "bang", age: 34 }
...     ]
... )
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5fd0fb5aaacce2fb8420b442"),
                ObjectId("5fd0fb5aaacce2fb8420b443"),
                ObjectId("5fd0fb5aaacce2fb8420b444")
        ]
}
>
```

READ operations

➜ Find() and Pretty()

➜ FindOne()

Find() and Pretty() : used to read elements

```
>
> db.flights.find().pretty()
{
        "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43e"),
        "departureAirport" : "IND",
        "arrivalAirport" : "LON",
        "aircraftNumber" : "Airbus A380",
        "distance" : 6000,
        "interContinental" : true
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b440"),
        "departureAirport" : "LON",
        "arrivalAirport" : "BOM",
        "aircraftNumber" : "Space X Crew-Dragon",
        "distance" : 6000,
        "interContinental" : true
}
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b441")
```

FindOne() : used to read only one instance.

```
> db.flights.find({capacity:"OPTIMAL"}).pretty()
> db.flights.find({capacity:"OPTIMAL"}).pretty()
>
>
> db.flights.find({Capacity :"OPTIMAL"}).pretty()
{
        "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false,
        "Capacity" : "OPTIMAL"
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false,
        "Capacity" : "OPTIMAL"
}
> db.flights.findOne({distance: {$gt : 2000}})
{
        "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false,
        "Capacity" : "OPTIMAL"
}
> db.flights.find({distance: {$gt : 2000}})
{ "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"), "departureAirport" : "CHN", "arrivalAirport" : "BGL", "aircraftNumber" : "Boeing A330", "distance" : 3000, "interContinental" : false, "Capacity" : "OPTIMAL" }
{ "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"), "departureAirport" : "CHN", "arrivalAirport" : "BGL", "aircraftNumber" : "Boeing A330", "distance" : 3000, "interContinental" : false, "Capacity" : "OPTIMAL" }
>
```

UPDATE operations

➔ UpdateOne()

➔ UpdateMany()

➔ ReplaceOne()

UpdateOne operation

Used to update an instance from the database.

```
> db.flights.updateOne({"distance":6000},{$set:{"features":"WIFI"}})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.flights.find().pretty()
{
        "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43e"),
        "departureAirport" : "IND",
        "arrivalAirport" : "LON",
        "aircraftNumber" : "Airbus A380",
        "distance" : 6000,
        "interContinental" : true,
        "features" : "WIFI"
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b440"),
        "departureAirport" : "LON",
        "arrivalAirport" : "BOM",
        "aircraftNumber" : "Space X Crew-Dragon",
        "distance" : 6000,
        "interContinental" : true
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b441"),
        "departureAirport" : "IXD",
        "arrivalAirport" : "JFK",
        "aircraftNumber" : "Virgin Galactic",
        "distance" : 6000,
        "interContinental" : true
}
>
```

UpdateMany():

Used to update multiple instance at once.

```
> db.flights.updateMany({},{$set:{"Capacity":"OPTIMAL"}})
{ "acknowledged" : true, "matchedCount" : 5, "modifiedCount" : 5 }
> db.flights.find()
{ "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"), "departureAirport" : "CHN", "arrivalAirport" : "BGL", "aircraftNumber" : "Boeing A330", "distance" : 3000, "interContinental" : false, "Capacity" : "OPTIMAL" }
{ "_id" : ObjectId("5fd0f0ecaacce2fb8420b43e"), "departureAirport" : "IND", "arrivalAirport" : "LON", "aircraftNumber" : "Airbus A380", "distance" : 6000, "interContinental" : true, "features" : "WIFI", "Capacity" : "OPTIMAL" }
{ "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"), "departureAirport" : "CHN", "arrivalAirport" : "BGL", "aircraftNumber" : "Boeing A330", "distance" : 3000, "interContinental" : false, "Capacity" : "OPTIMAL" }
{ "_id" : ObjectId("5fd0f0ecaacce2fb8420b440"), "departureAirport" : "LON", "arrivalAirport" : "BOM", "aircraftNumber" : "Space X Crew-Dragon", "distance" : 6000, "interContinental" : true, "Capacity" : "OPTIMAL" }
{ "_id" : ObjectId("5fd0f0ecaacce2fb8420b441"), "departureAirport" : "IXD", "arrivalAirport" : "JFK", "aircraftNumber" : "Virgin Galactic", "distance" : 6000, "interContinental" : true, "Capacity" : "OPTIMAL" }
> db.flights.find().pretty()
{
```

ReplaceOne(): Used to replace an instance.

```
>
> db.flights.replaceOne({"_id" : ObjectId("5fd0e45aaacce2fb8420b43d")},{"departureAirport":"LON",
... "arrivalAirport":"BOM",
... "aircraftNumber": "Space X Crew-Dragon",
... "distance": 6000,
... "interContinental": true})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.flights.find().pretty()
{
        "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"),
        "departureAirport" : "LON",
        "arrivalAirport" : "BOM",
        "aircraftNumber" : "Space X Crew-Dragon",
        "distance" : 6000,
        "interContinental" : true
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false,
        "Capacity" : "OPTIMAL"
}
>
```

DELETE operations

➔ DeleteOne()

➔ DeleteMany()

DeleteOne operation

Used to delete an instance from the database.

```
> db.flights.deleteOne({arrivalAirport :"JFK"})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.flights.find().pretty()
{
        "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false,
        "Capacity" : "OPTIMAL"
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43e"),
        "departureAirport" : "IND",
        "arrivalAirport" : "LON",
        "aircraftNumber" : "Airbus A380",
        "distance" : 6000,
        "interContinental" : true,
        "features" : "WIFI",
        "Capacity" : "OPTIMAL"
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false,
        "Capacity" : "OPTIMAL"
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b440"),
        "departureAirport" : "LON",
        "arrivalAirport" : "BOM",
        "aircraftNumber" : "Space X Crew-Dragon",
        "distance" : 6000,
        "interContinental" : true,
        "Capacity" : "OPTIMAL"
}
>
```

DeleteMany() :

Used to delete multiple instance at once.

```
}
> db.flights.deleteMany({"distance":6000})
{ "acknowledged" : true, "deletedCount" : 2 }
> db.flights.find()
{ "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"), "departureAirport" : "CHN", "arrivalAirport" : "BGL", "aircraftNumber" : "Boeing A330", "distance" : 3000, "interContinental" : false, "Capacity" : "OPTIMAL" }
{ "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"), "departureAirport" : "CHN", "arrivalAirport" : "BGL", "aircraftNumber" : "Boeing A330", "distance" : 3000, "interContinental" : false, "Capacity" : "OPTIMAL" }
> db.flights.find().pretty()
{
        "_id" : ObjectId("5fd0e45aaacce2fb8420b43d"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false,
        "Capacity" : "OPTIMAL"
}
{
        "_id" : ObjectId("5fd0f0ecaacce2fb8420b43f"),
        "departureAirport" : "CHN",
        "arrivalAirport" : "BGL",
        "aircraftNumber" : "Boeing A330",
        "distance" : 3000,
        "interContinental" : false,
        "Capacity" : "OPTIMAL"
}
```

Thank You!