



CHECKPOINT MONGODB

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

Please enter a MongoDB connection string (Default: mongodb://localhost/):

Current Mongosh Log ID: 64612ca3ecbde6e678ddba68

Connecting to: **mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.8.2**

Using MongoDB: 6.0.5

Using Mongosh: 1.8.2

For mongosh info see: <https://docs.mongodb.com/mongosh-shell/>

The server generated these startup warnings when booting

2023-05-11T11:23:45.422+01:00: 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()`

use contact

switched to db contact

contact> db.contactlist.insertMany([**{ lastname: "Ben Lahmer", firstname: "Fares", email: "fares@gmail.com", age:26},{lastname:"Kefi", firstname:"Seif", email:"kefi@gmail.com", age:15},{firstname:"Fatnassi",lastname:"Sarrah",email:"sarrah.f@gmail.com",age:40},{lastname:"Ben Yahia", firstname:"Rym",age:4},{lastname:"Cherif", firstname:"Sami",age:3}]]**)

acknowledged: **true**,

insertedIds: {

'0': ObjectId("64612ea0ecbde6e678ddba69"),

'1': ObjectId("64612ea0ecbde6e678ddba6a"),

'2': ObjectId("64612ea0ecbde6e678ddba6b"),

'3': ObjectId("64612ea0ecbde6e678ddba6c"),

'4': ObjectId("64612ea0ecbde6e678ddba6d")

- Display all the information about only one person using his ID.

```
contact> db.contactlist.find({_id : ObjectId("64612ea0ecbde6e678ddba6c")})
[
  {
    _id: ObjectId("64612ea0ecbde6e678ddba6c"),
    lastname: 'Ben Yahia',
    firstname: 'Rym',
    age: 4
  }
]
```

- Display all the contacts with an age >18.

```
contact> db.contactlist.find({age:{$gt:18}})
[
  {
    _id: ObjectId("64612ea0ecbde6e678ddba69"),
    lastname: 'Ben Lahmer',
    firstname: 'Fares',
    email: 'fares@gmail.com',
    age: 26
  },
  {
    _id: ObjectId("64612ea0ecbde6e678ddba6b"),
    firstname: 'Fatnassi',
    lastname: 'Sarraf',
    email: 'sarraf@gmail.com',
    age: 40
  }
]
```

- Display all the contacts with an age>18 and name containing "ah".

```
]
contact> db.contactlist.find({$and:[{age:{$gt:18}},{lastname:/ah/}]})
[
  {
    _id: ObjectId("64612ea0ecbde6e678ddba69"),
    lastname: 'Ben Lahmer',
    firstname: 'Fares',
    email: 'fares@gmail.com',
    age: 26
  }
]
```

- Change the contact's first name from "Kefi Seif" to "Kefi Anis".

```
]
contact> db.contactlist.updateOne({lastname:"Kefi"},{$set:{firstname:"Anis"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

- Delete the contacts that are aged under <5.

```
]
contact> db.contactlist.deleteMany({age:{$lt : 5}})
{ acknowledged: true, deletedCount: 2 }
```

- Display all of the contacts list.

```
contact> db.contactlist.find()
[
  {
    _id: ObjectId("64612ea0ecbde6e678ddba69"),
    lastname: 'Ben Lahmer',
    firstname: 'Fares',
    email: 'fares@gmail.com',
    age: 26
  },
  {
    _id: ObjectId("64612ea0ecbde6e678ddba6a"),
    lastname: 'Kefi',
    firstname: 'Anis',
    email: 'kefi@gmail.com',
    age: 15
  },
  {
    _id: ObjectId("64612ea0ecbde6e678ddba6b"),
    firstname: 'Fatnassi',
    lastname: 'Sarraf',
    email: 'sarraf@gmail.com',
    age: 40
  }
]
contact> _
```