

Mongod

# To initialize cmd line

mongo

# To start mongo cmdl

cls

show dbs

# To show pre present dbs

use <name>

# To start using and to create a db

db.dropDatabase()

# To delete the selected db

db.products.insertOne({name:"Max book",price: 12.99})

# To insert a data entry

db.products.find().pretty()

# To print the data of the db

db.flightData.insertOne({

... "departureAirport": "MUC",

... "arrivalAirport": "SFO",

... "aircraft": "Airbus A380",

... "distance": 12000,

... "intercontinental": true

... })

# To insert one data element

db.flightData.insertOne({"departureAirport":"TXL","arrivalAirport":"LHR"})

# To insert a data entry

db.flightData.insertOne({"\_id":"txl-lhr-1","departureAirport":"TXL","arrivalAirport":"LHR"})

# To insert a data entry with custom obj id

```
db.flightData.deleteOne({departureAirport:"TXL"})
```

# To delete a data entry with the given filter

```
db.flightData.updateOne({distance:12000},{ $set:{marker:'delete'}})
```

# To update an element with new row acc to the specified filter

```
db.flightData.updateMany({},{$set:{marker:'toDelete'}})
```

# To update all data entry so first field blank (all) and then add new row of marker

```
db.flightData.deleteMany({marker:"toDelete"})
```

# To delete all data elements that have marker row with the specified value

```
db.flightData.insertMany([  
  ... {  
  ...   "departureAirport": "MUC",  
  ...   "arrivalAirport": "SFO",  
  ...   "aircraft": "Airbus A380",  
  ...   "distance": 12000,  
  ...   "intercontinental": true  
  ... },  
  ... {  
  ...   "departureAirport": "LHR",  
  ...   "arrivalAirport": "TXL",  
  ...   "aircraft": "Airbus A320",  
  ...   "distance": 950,  
  ...   "intercontinental": false  
  ... }  
  ... ]  
  ... )
```

# To insert a data entry as list/array

```
db.flightData.find({intercontinental: true}).pretty()
```

# To print the data which has the mentioned filter

```
db.flightData.find({distance: {$gt : 900}}).pretty()
```

# To find and print all the data where the value of field meets the req filter condition(\$gt greater)

```
db.flightData.findOne({distance: {$gt : 900}})
```

# To find only one data element where it meets filter condition

# Pretty can't be used with one !

```
db.flightData.updateOne({"_id" : ObjectId("620f8c8135ebe6ae4c7dc7a1")},{ $set : {delayed:true}})
```

```
db.flightData.update({"_id" : ObjectId("620f8c8135ebe6ae4c7dc7a1")},{delayed:false})
```

#Overwriting issue

```
db.flightData.replaceOne({"_id" : ObjectId("620f8c8135ebe6ae4c7dc7a1")}, {"departureAirport":  
"MUC",
```

```
...   "arrivalAirport": "SFO",
```

```
...   "aircraft": "Airbus A380",
```

```
...   "distance": 12000,
```

```
...   "intercontinental": true})
```

# To update the data entry which has the given object\_id with the details

```
db.passengers.insertMany([
```

```
... {
```

```
...   "name": "Max SchwarzmueLLer",
```

```
...   "age": 29
```

```
... },  
  
... {  
...   "name": "Manu Lorenz",  
...   "age": 30  
... },  
  
... {  
...   "name": "Chris Hayton",  
...   "age": 35  
... },  
  
... {  
...   "name": "Sandeep Kumar",  
...   "age": 28  
... },  
  
... {  
...   "name": "Maria Jones",  
...   "age": 30  
... },  
  
... {  
...   "name": "Alexandra Maier",  
...   "age": 27  
... },  
  
... {  
...   "name": "Dr. Phil Evans",  
...   "age": 47  
... },  
  
... {  
...   "name": "Sandra Brugge",
```

```
...   "age": 33
... },
... {
...   "name": "Elisabeth Mayr",
...   "age": 29
... },
... {
...   "name": "Frank Cube",
...   "age": 41
... },
... {
...   "name": "Karandeep Alun",
...   "age": 48
... },
... {
...   "name": "Michaela Drayer",
...   "age": 39
... },
... {
...   "name": "Bernd Hoftstadt",
...   "age": 22
... },
... {
...   "name": "Scott Tolib",
...   "age": 44
... },
... {
```

```
...   "name": "Freddy Melfer",
...   "age": 41
... },
... {
...   "name": "Alexis Bohed",
...   "age": 35
... },
... {
...   "name": "Melanie Palace",
...   "age": 27
... },
... {
...   "name": "Armin Glutch",
...   "age": 35
... },
... {
...   "name": "Klaus Arber",
...   "age": 53
... },
... {
...   "name": "Albert Twostone",
...   "age": 68
... },
... {
...   "name": "Gordon Black",
...   "age": 38
... }
```

... ]

... )

# To insert a data entry as list/array

```
db.passengers.find().toArray()
```

# find only gives a marker so only 20 data restriction so we pass toArray to iterate and print all as array

```
db.passengers.find().forEach((passengerData) => {printjson(passengerData)})
```

# to get the data as Json

```
db.passengers.find({}, {name:1, _id:0}).pretty()
```

# to print a specific data row of a data element we pass as 1 to enable it other fields are default zero but to remove the id we have to manually assign 0

# projection

```
db.flightData.updateMany({}, {$set: {status: {description: "on-time", lastUpdated: "1 he ago"}}})
```

# embedded data

```
db.flightData.updateMany({}, {$set: {status: {description: "on-time", lastUpdated: "1 he ago", details: {responsible : 'Aditya'}}}}})
```

# embedded + nested data

```
db.passengers.updateOne({name: "Albert Twostone"}, {$set : {hobbies: ["sports", "cooking"]}})
```

# specific update with array data

```
db.passengers.find({name:"Albert Twostone"}).pretty()
```

# to print specific

```
db.passengers.findOne({name:"Albert Twostone"}).hobbies
```

# to print specific