

Lab 1 MongoDB Part - 1

```
C:\Users\student>mongoimport
2025-03-04T15:04:25.938+0530 no collection specified
2025-03-04T15:04:25.938+0530 using filename '' as collection
2025-03-04T15:04:25.938+0530 error validating settings: invalid collection name: collection name cannot be an empty string

C:\Users\student>mongoexport
2025-03-04T15:04:49.930+0530 must specify a collection
2025-03-04T15:04:49.931+0530 try 'mongoexport --help' for more information

C:\Users\student>mongoexport mongodb+srv://uzairobaid:uzairobaid123@cluster0.pdibg.mongodb.net/DBMS_DEMO --collection=Student --out C:\Users\student\Downloads\output.json
2025-03-04T15:11:46.757+0530 connected to: mongodb+srv://[**REDACTED**]@cluster0.pdibg.mongodb.net/DBMS_DEMO
2025-03-04T15:11:46.979+0530 exported 6 records
```

```
Atlas atlas-herlbh-shard-0 [primary] DBMS_DEMO> db.New_Student.find()
```

```
[
  {
    _id: ObjectId('67c6c2b14b7503e62cfa4215'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c2c04b7503e62cfa4216'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c2c74b7503e62cfa4217'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c2cd4b7503e62cfa4218'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'Abhinav@gmail.com'
  },
  {
    _id: ObjectId('67c6c3ac4b7503e62cfa4219'),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  },
  {
    _id: ObjectId('67c6c27b4b7503e62cfa4214'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  }
]
```

I. CREATE DATABASE IN MONGODB.

use myDB;

Confirm the existence of your database

db;

To list all databases

show dbs;

II. CRUD (CREATE, READ, UPDATE, DELETE) OPERATIONS

1. To create a collection by the name "Student". Let us take a look at the collection list prior to the creation of the new collection "Student".

db.createCollection("Student");

2. To drop a collection by the name "Student".

db.Student.drop();

3. Create a collection by the name "Students" and store the following data in it.

db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"InternetSurfing"});

4. Insert the document for "AryanDavid" in to the Students collection only if it does not already exist in the collection.

db.Student.update({_id:3,StudName:"AryanDavid",Grade:"VII"},{\$set:{Hobbies:"Skating"}},{upsert:true});

5. FIND METHOD

A. To search for documents from the "Students" collection based on certain search criteria.

db.Student.find({StudName:"Aryan David"});

B. To display only the StudName and Grade from all the documents of the Students collection. The identifier _id should be suppressed and NOT displayed.

db.Student.find({}, {StudName:1,Grade:1,_id:0});

C. To find those documents where the Grade is set to 'VII'

db.Student.find({Grade:{ \$eq:'VII'}}).pretty();

D. To find those documents from the Students collection where the Hobbies is set to either 'Chess' or is set to 'Skating'.

db.Student.find({Hobbies :{ \$in: ['Chess','Skating']}}).pretty ();

E. To find documents from the Students collection where the StudName begins with "M".

db.Student.find({StudName:/^M/}).pretty();

F. To find documents from the Students collection where the StudName has an "e" in any position.

db.Student.find({StudName:/e/}).pretty();

G. To find the number of documents in the Students collection.

db.Student.count();

H. To sort the documents from the Students collection in the descending order of StudName.

db.Student.find().sort({StudName:-1}).pretty();

III. Import data from a CSV file

Given a CSV file "sample.txt" in the D:drive, import the file into the MongoDB collection, "SampleJSON". The collection is in the database "test".

mongoimport --db Student --collection airlines --type csv --headerline --file /home/hduser/Desktop/airline.csv

IV. Export data to a CSV file

This command used at the command prompt exports MongoDB JSON documents from “Customers” collection in the “test” database into a CSV file “Output.txt” in the D:drive.

mongoexport --host localhost --db Student --collection airlines --csv --out /home/hduser/Desktop/output.txt --fields “Year”, “Quarter”

V. Save Method :

Save() method will insert a new document, if the document with the _id does not exist. If it exists it will replace the existing document:

```
db.Students.save({StudName:“Vamsi”, Grade:“VI”})
```

VI. Add a new field to existing Document:

```
db.Students.update({_id:4},{ $set:{Location:“Network”}})
```

VII. Remove the field in an existing Document

```
db.Students.update({_id:4},{ $unset:{Location:“Network”}})
```

VIII. Finding Document based on search criteria suppressing few fields

```
db.Student.find({_id:1},{StudName:1,Grade:1,_id:0});
```

To find those documents where the Grade is not set to ‘VII’

```
db.Student.find({Grade:{ $ne:‘VII’}}).pretty();
```

To find documents from the Students collection where the StudName ends with s.

```
db.Student.find({StudName:/s$/}).pretty();
```

IX. to set a particular field value to NULL

```
db.Students.update({_id:3},{ $set:{Location:null}})
```

X. Count the number of documents in Student Collections

```
db.Students.count()
```

XI. Count the number of documents in Student Collections with grade :VII

```
db.Students.count({Grade:“VII”})
```

retrieve first 3 documents

```
db.Students.find({Grade:“VII”}).limit(3).pretty();
```

Sort the document in Ascending order

```
db.Students.find().sort({StudName:1}).pretty();
```

to Skip the 1st two documents from the Students Collections

```
db.Students.find().skip(2).pretty()
```

XII. Create a collection by name “food” and add to each document add a “fruits” array

```
db.food.insert( { _id:1, fruits:['grapes','mango','apple'] } )
```

```
db.food.insert( { _id:2, fruits:['grapes','mango','cherry'] } )
```

```
db.food.insert( { _id:3, fruits:['banana','mango'] } )
```

To find those documents from the “food” collection which has the “fruits array” constitute of “grapes”, “mango” and “apple”.

```
db.food.find ( {fruits: ['grapes','mango','apple'] } ). pretty().
```

To find in “fruits” array having “mango” in the first index position.

```
db.food.find ( {'fruits.1':'grapes' } )
```

To find those documents from the “food” collection where the size of the array is two.

```
db.food.find ( {“fruits”: {$size:2}} )
```

To find the document with a particular id and display the first two elements from the array “fruits”

```
db.food.find({_id:1},{“fruits”:$slice:2}})
```

To find all the documents from the food collection which have elements mango and grapes in the array “fruits”

```
db.food.find({fruits:{$all:[“mango”,“grapes”]}})
```

update on Array:

using particular id replace the element present in the 1st index position of the fruits array with apple

```
db.food.update({_id:3},{set: {'fruits.1': 'apple'}})
```

insert new key value pairs in the fruits array

```
db.food.update({_id:2},{push: {price: {grapes:80,mango:200,cherry:100}}})
```

XII. Aggregate Function :

Create a collection Customers with fields custID, AcctBal, AcctType.

Now group on “custID” and compute the sum of “AccBal”.

```
db.Customers.aggregate ( {$group : { _id : “$custID”,TotAccBal : {$sum:”$AccBal”} } } );
```

match on AcctType:”S” then group on “CustID” and compute the sum of “AccBal”.

```
db.Customers.aggregate ( {$match: {AcctType:”S”}},{$group : { _id : “$custID”,TotAccBal : {$sum:”$AccBal”} } } );
```

match on AcctType:”S” then group on “CustID” and compute the sum of “AccBal” and total balance greater than 1200.

```
db.Customers.aggregate ( {$match: {AcctType:”S”}},{$group : { _id : “$custID”,TotAccBal : {$sum:”$AccBal”} } },  
{$match: {TotAccBal: {$gt:1200}}});
```

```
C:\Users\student>mongoimport mongodb+srv://uzairobaid:uzairobaid123@cluster0.pdibg.mongodb.net/DBMS_DEMO --collection=New_Student --type json --file C:\Users\student\Downloads\output.json
2025-03-04T15:19:31.071+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.pdibg.mongodb.net/DBMS_DEMO
2025-03-04T15:19:31.168+0530    6 document(s) imported successfully. 0 document(s) failed to import.
```

```
C:\Users\student>mongoimport mongodb+srv://uzairobaid:uzairobaid123@cluster0.pdibg.mongodb.net/DBMS_DEMO --collection=New_Student --type json --file C:\Users\student\Downloads\output.json
2025-03-04T15:19:31.071+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.pdibg.mongodb.net/DBMS_DEMO
2025-03-04T15:19:31.168+0530    6 document(s) imported successfully. 0 document(s) failed to import.
```