LAB 1 - 1BM22CS157 - BDA MongoDB commands

```
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.insert({_id:"1",StudName:"MichelleJacintha",Grade:"VIII",Hobbies:"InternetSurfing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': '1' } }
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.insertMany([{_id:"2",StudName:"johnCanny",grade:"V",Hobbies:"blogging"},{_id:"3",StudName:"JaneSmith",grade:"V",Hobbies:"graphicDesign"}]);
{ acknowledged: true, insertedIds: { '0': '2', '1': '3' } }
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.find()
[
  {
    _id: '1',
    StudName: 'MichelleJacintha',
    Grade: 'VIII',
    Hobbies: 'InternetSurfing'
  },
  { _id: '2', StudName: 'johnCanny', grade: 'V', Hobbies: 'blogging' },
  {
    _id: '3',
    StudName: 'JaneSmith',
    grade: 'V',
    Hobbies: 'graphicDesign'
  }
]
Atlas atlas-x4w7ev-shard-0 [primary] myDB>
```

```
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.insert({_id:"1",StudName:"MichelleJacintha",Grad
e:"VIII",Hobbies:"InternetSurfing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': '1' } }
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.insertMany([{_id:"2",StudName:"johnCanny",grade:
"V",Hobbies:"blogging"},{_id:"3",StudName:"JaneSmith",grade:"V",Hobbies:"graphicDesign"}]);
{ acknowledged: true, insertedIds: { '0': '2', '1': '3' } }
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.find()
[
  {
    _id: '1',
    StudName: 'MichelleJacintha',
    Grade: 'VIII',
    Hobbies: 'InternetSurfing'
  },
  { _id: '2', StudName: 'johnCanny', grade: 'V', Hobbies: 'blogging' },
  {
    _id: '3',
    StudName: 'JaneSmith',
    grade: 'V',
    Hobbies: 'graphicDesign'
  }
]
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.find({},{StudName:1,Grade:1,_id:0});
[
  { StudName: 'MichelleJacintha', Grade: 'VIII' },
  { StudName: 'johnCanny' },
  { StudName: 'JaneSmith' }
]
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.find({},{StudName:1,Grade:"1",_id:0});
[
  { StudName: 'MichelleJacintha', Grade: '1' },
  { StudName: 'johnCanny', Grade: '1' },
  { StudName: 'JaneSmith', Grade: '1' }
]
Atlas atlas-x4w7ev-shard-0 [primary] myDB> db.Student.find({},{StudName:1,Grade:1,_id:0});
[
  { StudName: 'MichelleJacintha', Grade: 'VIII' },
  { StudName: 'johnCanny' },
  { StudName: 'JaneSmith' }
]
Atlas atlas-x4w7ev-shard-0 [primary] myDB> D
```

WORKING WITH MONGODB

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"); => *sql equivalent* **CREATE TABLE 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. However, if it is already present in the collection, then update the document with new values. (Update his Hobbies from "Skating" to "Chess".) Use "Update else insert" (if there is an existing document, it will attempt to update it, if there is no existing document then it will insert it).

   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 StudNamehas 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"

Sort the document in Ascending order

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

Note:

for desending order : db.Students.find().sort({StudName:-1}).pretty();

to Skip the 1 st 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 documets 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 1 st 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}}})

Note: perform query operations using - pop, addToSet, pullAll and pull

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}}});

Assignment:

Creation of Cursor:

Create Collection "Alphabets"

Insert Documents with fields "_id" and "alphabet" use

cursor to iterate through the "Alphabets" Collection.

```
PS C:\Users\student>  mongoexport mongodb+srv://shuraihshaikhcs22:izJPn5Of32Zwqvqv@cluster0.pevls.mongodb.net/dbms_demo
 --collection=Student --out C:\\Users\\student\\Desktop\\out.json
2025-03-04T15:20:09.598+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.pevls.mongodb.net/dbms_demo
2025-03-04T15:20:10.128+0530    exported 6 records
PS C:\Users\student>  mongoimport mongodb+srv://shuraihshaikhcs22:izJPn5Of32Zwqvqv@cluster0.pevls.mongodb.net/dbms_demo
 --collection=Student --type json --file C:\\Users\\student\\Desktop\\out.json
2025-03-04T15:22:34.696+0530    connected to: mongodb+srv://[**REDACTED**]@cluster0.pevls.mongodb.net/dbms_demo
2025-03-04T15:22:34.830+0530    6 document(s) imported successfully. 0 document(s) failed to import.
PS C:\Users\student> |
```