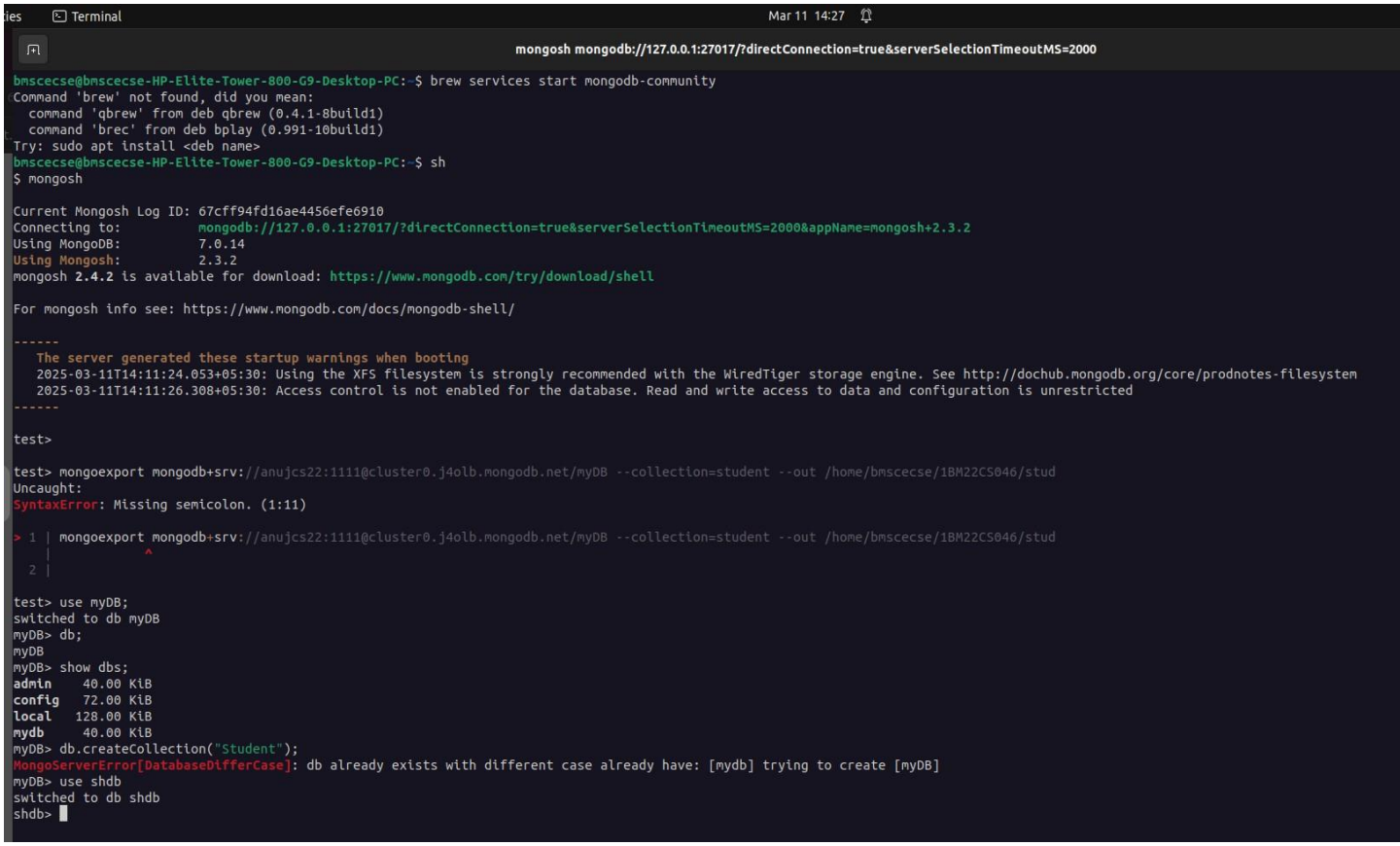
LAB 2 - 1BM22CS066 - BDA MongoDB commands

Mongodb Lab Exercise

1.Perform the following DB operations using MongoDB.
1. Create a collection by name Customers with the following attributes.
Cust_id, Acc_Bal, Acc_Type
2. Insert at least 5 values into the table
3. Write a query to display those records whose total account balance is greater than 1200 of account type 'Z' for each customer_id.
4. Determine Minimum and Maximum account balance for each customer_i

2. You are developing an e-commerce platform where users can browse and purchase products. Each product has a unique identifier, a name, a category, a price, and available quantity. Additionally, users can add products to their cart and place orders. Design a MongoDB schema to efficiently handle product information, user carts, and orders.
Query to
Retrieve All Products.
Retrieve Products in a Specific Category (e.g., Electronics).
Retrieve Products with Quantity Greater Than 0.
Retrieve Products Sorted by Price in Ascending Order.
Retrieve Products with Price Less Than or Equal to $100.
Retrieve Products Added to a User's Cart (User with ID "789ghi...")
Retrieve Orders Placed by a User (User with ID "123abc...")
Retrieve Total Price of Orders Placed by a User (User with ID "123abc...")
Additional Aggregation queries based on Assignment-3 design:
1 .Calculate Total Number of Products in Each Category.
2. Calculate Total Price of Products in Each Category.
3. Find Average Price of Products.
4. Find Products with Quantity Less Than 10.
5. Sort Products by Price in Descending Order.
6. Calculate Total Price of Orders Placed by Each User.
7. Find Users with the Highest Total Price of Orders.
8. Find Average Total Price of Orders.

LAB 2 - 1BM22CS066 - BDA MongoDB commands



LAB 2 - 1BM22CS066 - BDA MongoDB commands

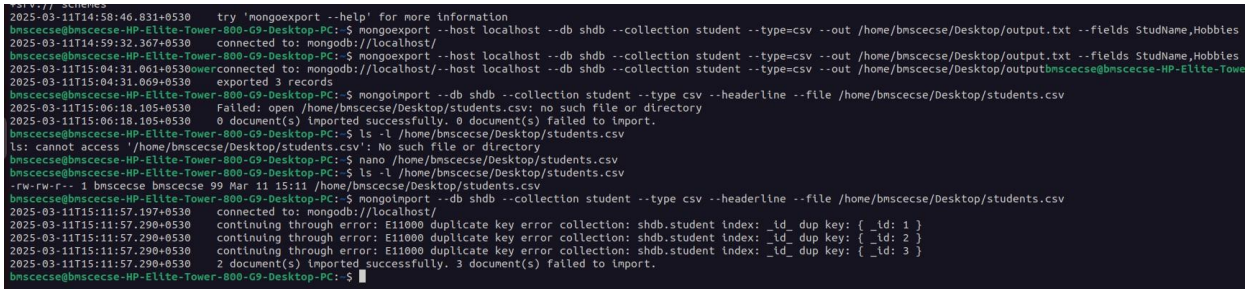
```
myDB> use shdb
switched to db shdb
shdb> db.createCollection("Student");
{ ok: 1 }
shdb> db.Student.drop();
true
shdb> db.createCollection("Student");
{ ok: 1 }
shdb> db.Student.insert({_id:1, StudName:"MichelleJacintha", Grade:"VII", Hobbies:"InternetSurfing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 1 } }
shdb> db.Student.update(
...   {_id:3, StudName:"AryanDavid", Grade:"VII"},
...   {$set:{Hobbies:"Skating"}},
...   {upsert:true}
... );
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: 3,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
shdb> db.Student.find({StudName:"Aryan David"});

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

shdb> db.Student.find({}, {StudName:1, Grade:1, _id:0});
[
  { StudName: 'MichelleJacintha', Grade: 'VII' },
  { Grade: 'VII', StudName: 'AryanDavid' }
]
shdb> db.Student.find({StudName:"Aryan David"});

shdb> db.Student.find();
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 3, Grade: 'VII', StudName: 'AryanDavid', Hobbies: 'Skating' }
]
shdb> db.Student.find({StudName:"Arya David"});

shdb> db.Student.find({StudName:"AryaDavid"});

shdb> db.Student.find({}, {StudName:1, Grade:1, _id:0});
[
  { StudName: 'MichelleJacintha', Grade: 'VII' },
  { Grade: 'VII', StudName: 'AryanDavid' }
]
shdb>
```

LAB 2 - 1BM22CS066 - BDA MongoDB commands

```
shdb> db.Student.find({Grade:{$eq:'VII'}}).pretty();
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 3, Grade: 'VII', StudName: 'AryanDavid', Hobbies: 'Skating' }
]
shdb> db.Student.find({Hobbies: { $in: ['Chess','Skating'] }}).pretty();
[
  { _id: 3, Grade: 'VII', StudName: 'AryanDavid', Hobbies: 'Skating' }
]
shdb> db.Student.find({StudName:/^M/}).pretty();
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
shdb> db.Student.find({StudName:/e/}).pretty();
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  }
]
shdb> db.Student.count();
DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.
2
shdb> db.Student.find().sort({StudName:-1}).pretty();
[
  {
    _id: 1,
    StudName: 'MichelleJacintha',
    Grade: 'VII',
    Hobbies: 'InternetSurfing'
  },
  { _id: 3, Grade: 'VII', StudName: 'AryanDavid', Hobbies: 'Skating' }
]
shdb>
```

```
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db shdb --collection student --csv --out /home/hduser/Desktop/output.txt --fields "Year","Quarter"
2025-03-11T14:49:35.969+0530    csv flag is deprecated; please use --type=csv instead
2025-03-11T14:49:35.974+0530    connected to: mongodb://localhost/
2025-03-11T14:49:35.974+0530    error opening output stream: mkdir /home/hduser: permission denied
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ pwd
/home/bmscecse
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db shdb --collection student --csv --out /home/bmscecse/Desktop/output.txt --fields "Year","Quarter"
2025-03-11T14:51:23.863+0530    csv flag is deprecated; please use --type=csv instead
2025-03-11T14:51:23.868+0530    connected to: mongodb://localhost/
2025-03-11T14:51:23.869+0530    exported 0 records
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db shdb --collection student --type=csv --out /home/bmscecse/Desktop/output.txt
2025-03-11T14:55:42.185+0530    connected to: mongodb://localhost/
2025-03-11T14:55:42.186+0530    exported 0 records
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db shdb --collection student --csv --out /home/bmscecse/Desktop/output.txt --fields "StudName", "Hobbies"
2025-03-11T14:58:46.831+0530    error parsing command line options: error parsing positional arguments: provide only one MongoDB connection string. Connection strings must begin with mongodb:// or mongodb
+srv:// schemes
2025-03-11T14:58:46.831+0530    try 'mongoexport --help' for more information
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db shdb --collection student --type=csv --out /home/bmscecse/Desktop/output.txt --fields StudName,Hobbies
2025-03-11T14:59:32.367+0530    connected to: mongodb://localhost/
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db shdb --collection student --type=csv --out /home/bmscecse/Desktop/output.txt --fields StudName,Hobbies
2025-03-11T15:04:31.061+0530owerconnected to: mongodb://localhost/--host localhost --db shdb --collection student --type=csv --out /home/bmscecse/Desktop/outputbmscecse@bmscecse-HP-Elite-Tower-800-G9-Desk
2025-03-11T15:04:31.069+0530    exported 3 records
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoimport --db shdb --collection student --type csv --headerline --file /home/bmscecse/Desktop/students.csv
2025-03-11T15:06:18.105+0530    Failed: open /home/bmscecse/Desktop/students.csv: no such file or directory
2025-03-11T15:06:18.105+0530    0 document(s) imported successfully. 0 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls -l /home/bmscecse/Desktop/students.csv
ls: cannot access '/home/bmscecse/Desktop/students.csv': No such file or directory
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ nano /home/bmscecse/Desktop/students.csv
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls -l /home/bmscecse/Desktop/students.csv
-rw-rw-r-- 1 bmscecse bmscecse 99 Mar 11 15:11 /home/bmscecse/Desktop/students.csv
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoimport --db shdb --collection student --type csv --headerline --file /home/bmscecse/Desktop/students.csv
2025-03-11T15:11:57.197+0530    connected to: mongodb://localhost/
2025-03-11T15:11:57.290+0530    continuing through error: E11000 duplicate key error collection: shdb.student index: _id_ dup key: { _id: 1 }
2025-03-11T15:11:57.290+0530    continuing through error: E11000 duplicate key error collection: shdb.student index: _id_ dup key: { _id: 2 }
2025-03-11T15:11:57.290+0530    continuing through error: E11000 duplicate key error collection: shdb.student index: _id_ dup key: { _id: 3 }
2025-03-11T15:11:57.290+0530    2 document(s) imported successfully. 3 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

LAB 2 - 1BM22CS066 - BDA MongoDB commands



```
2025-03-11T14:58:46.831+0530    try 'mongoexport --help' for more information
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db shdb --collection student --type=csv --out /home/bmscecse/Desktop/output.txt --fields StudName,Hobbies
2025-03-11T14:59:32.367+0530    connected to: mongodb://localhost/
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoexport --host localhost --db shdb --collection student --type=csv --out /home/bmscecse/Desktop/output.txt --fields StudName,Hobbies
2025-03-11T15:04:31.061+0530connected to: mongodb://localhost/--host localhost --db shdb --collection student --type=csv --out /home/bmscecse/Desktop/outputbmscecse@bmscecse-HP-Elite-Towe
2025-03-11T15:04:31.069+0530    exported 3 records
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoimport --db shdb --collection student --type csv --headerline --file /home/bmscecse/Desktop/students.csv
2025-03-11T15:06:18.105+0530    Failed: open /home/bmscecse/Desktop/students.csv: no such file or directory
2025-03-11T15:06:18.105+0530    0 document(s) imported successfully. 0 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls -l /home/bmscecse/Desktop/students.csv
ls: cannot access '/home/bmscecse/Desktop/students.csv': No such file or directory
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ nano /home/bmscecse/Desktop/students.csv
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ ls -l /home/bmscecse/Desktop/students.csv
-rw-rw-r-- 1 bmscecse bmscecse 99 Mar 11 15:11 /home/bmscecse/Desktop/students.csv
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$ mongoimport --db shdb --collection student --type csv --headerline --file /home/bmscecse/Desktop/students.csv
2025-03-11T15:11:57.197+0530    connected to: mongodb://localhost/
2025-03-11T15:11:57.290+0530    continuing through error: E11000 duplicate key error collection: shdb.student index: _id_ dup key: { _id: 1 }
2025-03-11T15:11:57.290+0530    continuing through error: E11000 duplicate key error collection: shdb.student index: _id_ dup key: { _id: 2 }
2025-03-11T15:11:57.290+0530    continuing through error: E11000 duplicate key error collection: shdb.student index: _id_ dup key: { _id: 3 }
2025-03-11T15:11:57.290+0530    2 document(s) imported successfully. 3 document(s) failed to import.
bmscecse@bmscecse-HP-Elite-Tower-800-G9-Desktop-PC:~$
```

```
TypeError: db.Students.save is not a function
test> db.Students.update({_id:4}, {$set:{Location:"Network"}});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Students.update({_id:4}, {$unset:{Location:"Network"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Students.update({_id:4}, {$unset:{Location:"Network"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Students.update({_id:3}, {$set:{Location:null}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Students.find({Grade:"VII"}).limit(3).pretty();

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

test> db.food.insert({_id:1, fruits:['grapes','mango','apple']});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 1 } }
test> db.food.insert({_id:2, fruits:['grapes','mango','cherry']});
{ acknowledged: true, insertedIds: { '0': 2 } }
test> db.food.insert({_id:3, fruits:['banana','mango']});
{ acknowledged: true, insertedIds: { '0': 3 } }
test> db.food.find({fruits: ['grapes','mango','apple']}).pretty();
[ { _id: 1, fruits: [ 'grapes', 'mango', 'apple' ] } ]
test> db.food.find({'fruits.1':'grapes'});

test> db.food.find({"fruits": {$size:3}});
```

LAB 2 - 1BM22CS066 - BDA MongoDB commands

```
test> db.food.find({"fruits": {$size:2}});
[ { _id: 3, fruits: [ 'banana', 'mango' ] } ]
test> db.food.find({_id:1},{"fruits":{$slice:2}});
[ { _id: 1, fruits: [ 'grapes', 'mango' ] } ]
test> db.food.update({_id:3}, {$set: {'fruits.1':'apple'}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.food.update({_id:2}, {$push: {price:{grapes:80,mango:200,cherry:100}}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.food.update({_id:3}, {$set: {'fruits.1':'apple'}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Customers.aggregate([{ $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } }]);

test> db.Customers.aggregate([
...    { $match:{AcctType:"S"} },
...    { $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } }
... ]);

test> db.Customers.aggregate([
...    { $match:{AcctType:"S"} },
...    { $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } },
...    { $match:{TotAccBal:{$gt:1200}}}
... ]);

test> db.Alphabets.insertMany([{_id:1, alphabet:"A"}, {_id:2, alphabet:"B"}, {_id:3, alphabet:"C"}]);
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3 } }
test> var myCursor = db.Alphabets.find();
```

LAB 2 - 1BM22CS066 - BDA MongoDB commands

```
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
test> db.food.update({_id:3}, {$set: {'fruits.1':'apple'}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 0,
  upsertedCount: 0
}
test> db.Customers.aggregate([{ $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } }]);

test> db.Customers.aggregate([
...    { $match:{AcctType:"S"} },
...    { $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } }
... ]);

test> db.Customers.aggregate([
...    { $match:{AcctType:"S"} },
...    { $group : { _id : "$custID", TotAccBal : {$sum:"$AcctBal"} } },
...    { $match:{TotAccBal:{$gt:1200}}}
... ]);

test> db.Alphabets.insertMany([{_id:1, alphabet:"A"}, {_id:2, alphabet:"B"}, {_id:3, alphabet:"C"}]);
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3 } }
test> var myCursor = db.Alphabets.find();

test> while (myCursor.hasNext()) {
...      printjson(myCursor.next());
... }
{
  _id: 1,
  alphabet: 'A'
}
{
  _id: 2,
  alphabet: 'B'
}
{
  _id: 3,
  alphabet: 'C'
}

test> show dbs;
admin    40.00 KiB
config  108.00 KiB
local   128.00 KiB
mydb     40.00 KiB
shdb    112.00 KiB
test     96.00 KiB
test>
```