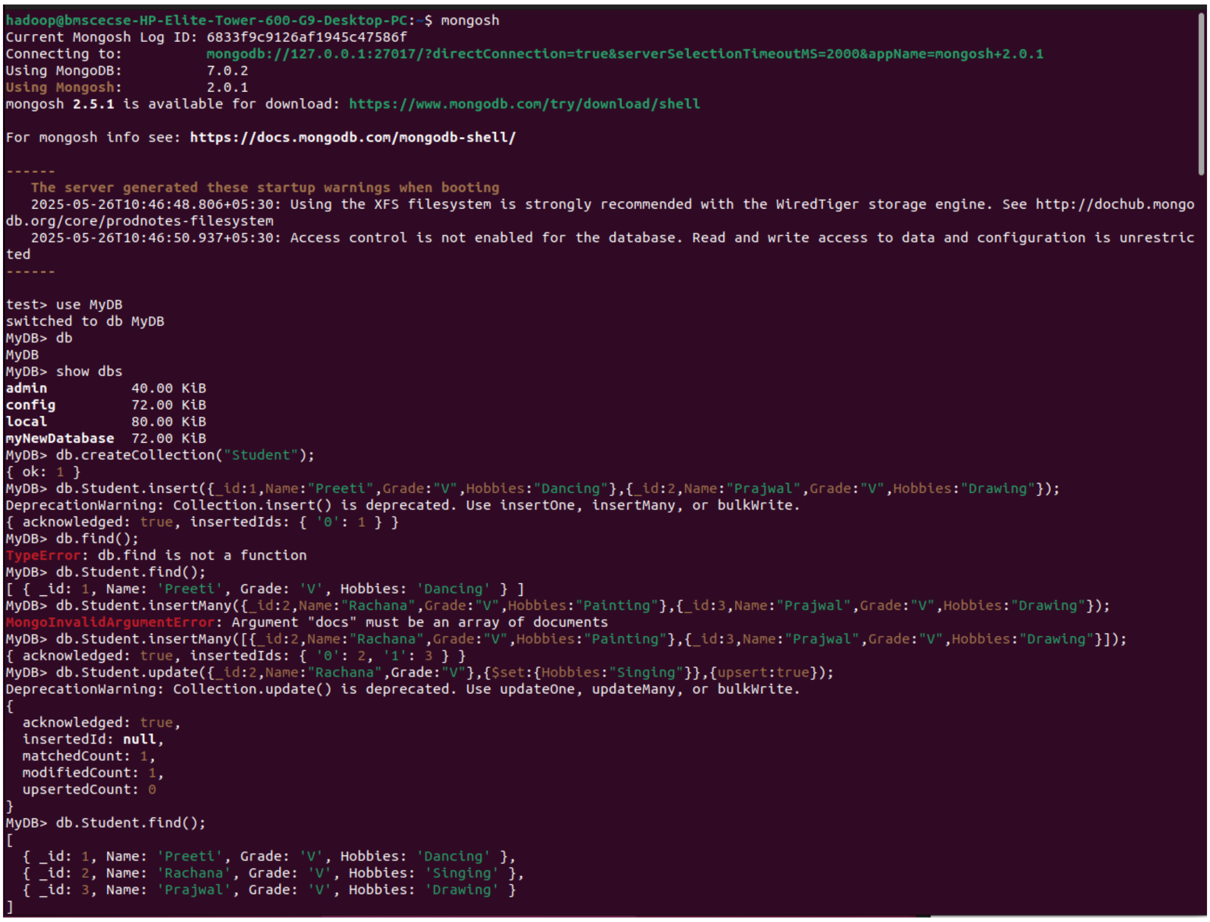
MongoDB- CRUD Operations Demonstration (Practice and Self Study)

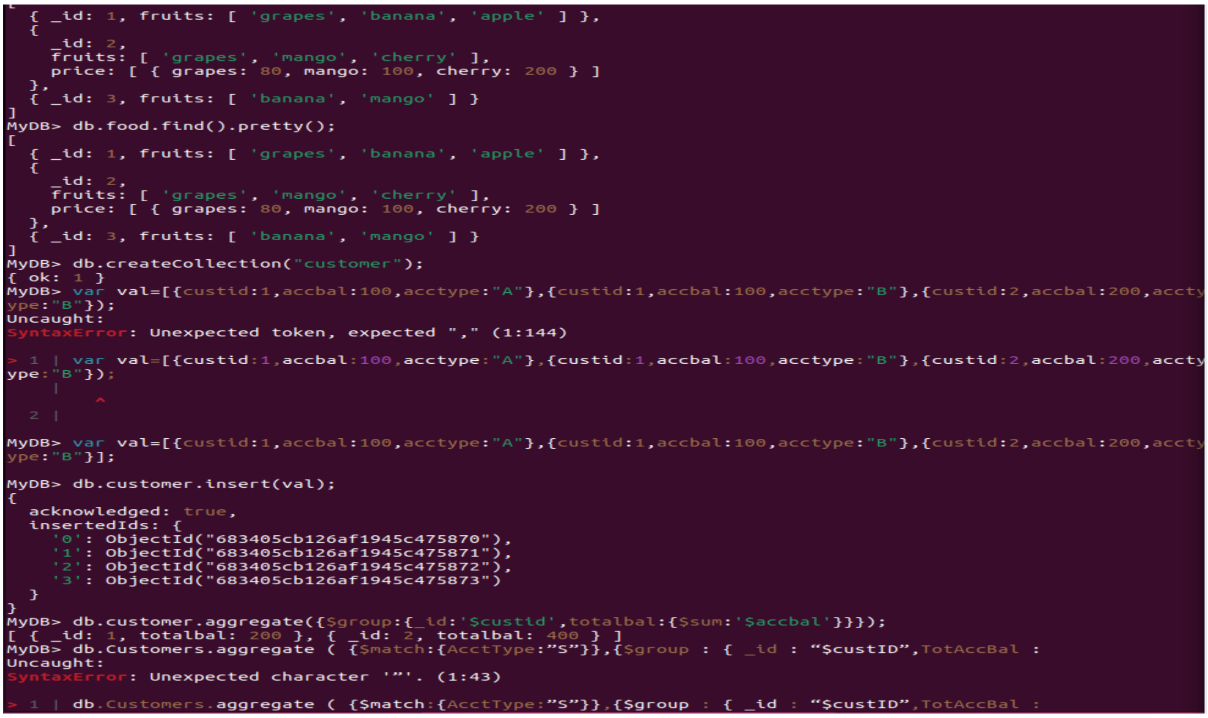


```
hadoop@bmscecse-HP-Elite-Tower-600-G9-Desktop-PC:~$ mongosh
Current Mongosh Log ID: 6833f9c9126af1945c47586f
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.1
Using MongoDB:          7.0.2
Using Mongosh:          2.0.1
mongosh 2.5.1 is available for download: https://www.mongodb.com/try/download/shell

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

------
   The server generated these startup warnings when booting
   2025-05-26T10:46:48.806+05:30: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongo
db.org/core/prodnotes-filesystem
   2025-05-26T10:46:50.937+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestric
ted
------

test> use MyDB
switched to db MyDB
MyDB> db
MyDB
MyDB> show dbs
admin          40.00 KiB
config         72.00 KiB
local          80.00 KiB
myNewDatabase  72.00 KiB
MyDB> db.createCollection("Student");
{ ok: 1 }
MyDB> db.Student.insert({_id:1,Name:"Preeti",Grade:"V",Hobbies:"Dancing"},{_id:2,Name:"Prajwal",Grade:"V",Hobbies:"Drawing"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{ acknowledged: true, insertedIds: { '0': 1 } }
MyDB> db.find();
TypeError: db.find is not a function
MyDB> db.Student.find();
[ { _id: 1, Name: 'Preeti', Grade: 'V', Hobbies: 'Dancing' } ]
MyDB> db.Student.insertMany({_id:2,Name:"Rachana",Grade:"V",Hobbies:"Painting"},{_id:3,Name:"Prajwal",Grade:"V",Hobbies:"Drawing"});
MongoInvalidArgumentError: Argument "docs" must be an array of documents
MyDB> db.Student.insertMany([{_id:2,Name:"Rachana",Grade:"V",Hobbies:"Painting"},{_id:3,Name:"Prajwal",Grade:"V",Hobbies:"Drawing"}]);
{ acknowledged: true, insertedIds: { '0': 2, '1': 3 } }
MyDB> db.Student.update({_id:2,Name:"Rachana",Grade:"V"},{$set:{Hobbies:"Singing"}},{upsert:true});
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
MyDB> db.Student.find();
[
  { _id: 1, Name: 'Preeti', Grade: 'V', Hobbies: 'Dancing' },
  { _id: 2, Name: 'Rachana', Grade: 'V', Hobbies: 'Singing' },
  { _id: 3, Name: 'Prajwal', Grade: 'V', Hobbies: 'Drawing' }
]
```



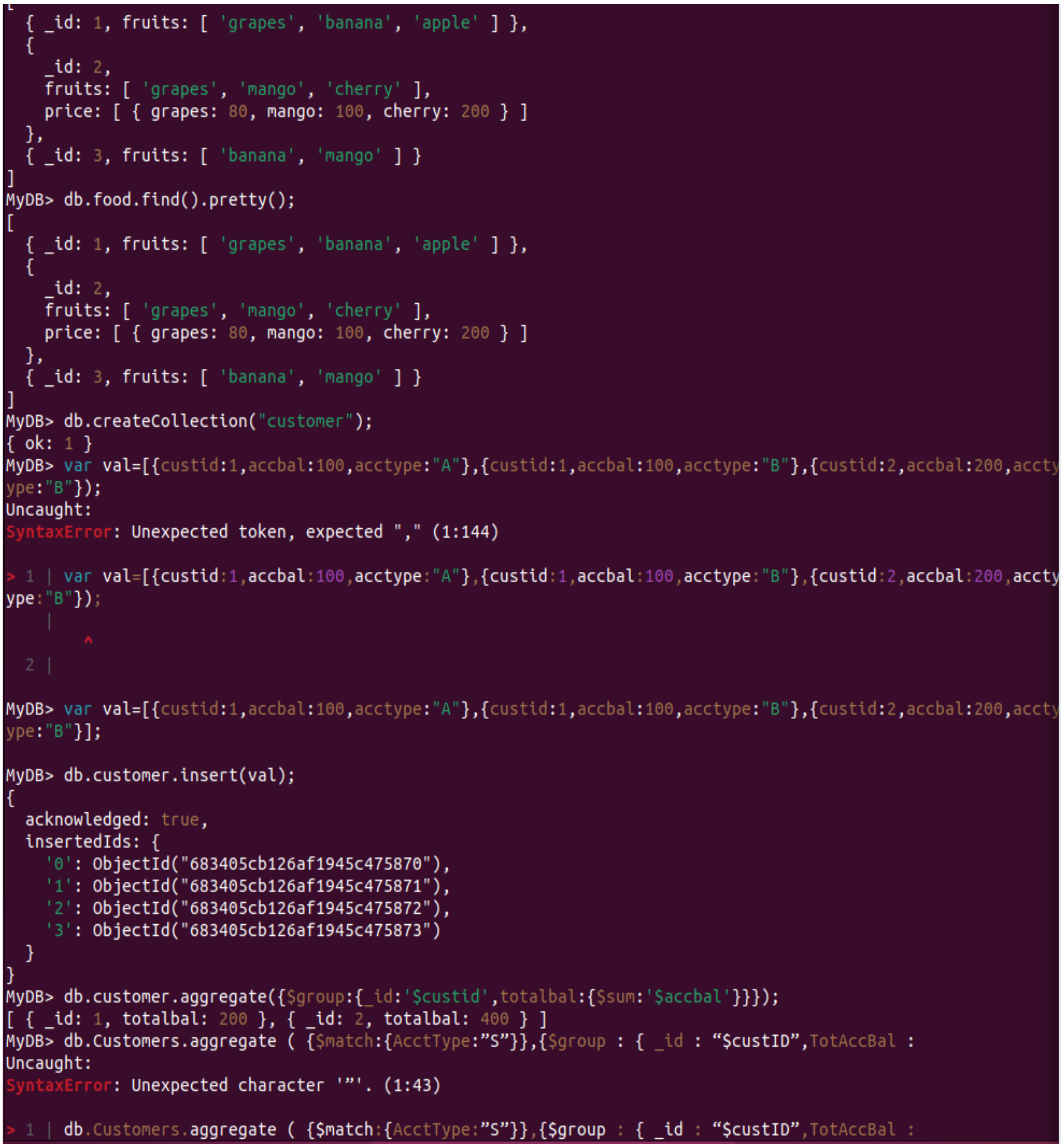
```
  { _id: 1, fruits: [ 'grapes', 'banana', 'apple' ] },
  {
    _id: 2,
    fruits: [ 'grapes', 'mango', 'cherry' ],
    price: [ { grapes: 80, mango: 100, cherry: 200 } ]
  },
  { _id: 3, fruits: [ 'banana', 'mango' ] }
]
MyDB> db.food.find().pretty();
[
  { _id: 1, fruits: [ 'grapes', 'banana', 'apple' ] },
  {
    _id: 2,
    fruits: [ 'grapes', 'mango', 'cherry' ],
    price: [ { grapes: 80, mango: 100, cherry: 200 } ]
  },
  { _id: 3, fruits: [ 'banana', 'mango' ] }
]
MyDB> db.createCollection("customer");
{ ok: 1 }
MyDB> var val=[{custid:1,accbal:100,acctype:"A"},{custid:1,accbal:100,acctype:"B"},{custid:2,accbal:200,accty
ype:"B"});
Uncaught:
SyntaxError: Unexpected token, expected "," (1:144)

> 1 | var val=[{custid:1,accbal:100,acctype:"A"},{custid:1,accbal:100,acctype:"B"},{custid:2,accbal:200,accty
ype:"B"});
    |
         ^
  2 |

MyDB> var val=[{custid:1,accbal:100,acctype:"A"},{custid:1,accbal:100,acctype:"B"},{custid:2,accbal:200,accty
ype:"B"}];

MyDB> db.customer.insert(val);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("683405cb126af1945c475870"),
    '1': ObjectId("683405cb126af1945c475871"),
    '2': ObjectId("683405cb126af1945c475872"),
    '3': ObjectId("683405cb126af1945c475873")
  }
}
MyDB> db.customer.aggregate({$group:{_id:'$custid',totalbal:{$sum:'$accbal'}}});
[ { _id: 1, totalbal: 200 }, { _id: 2, totalbal: 400 } ]
MyDB> db.Customers.aggregate ( {$match:{AcctType:”S”}},{$group : { _id : “$custID”,TotAccBal :
Uncaught:
SyntaxError: Unexpected character '”'. (1:43)

> 1 | db.Customers.aggregate ( {$match:{AcctType:”S”}},{$group : { _id : “$custID”,TotAccBal :
```



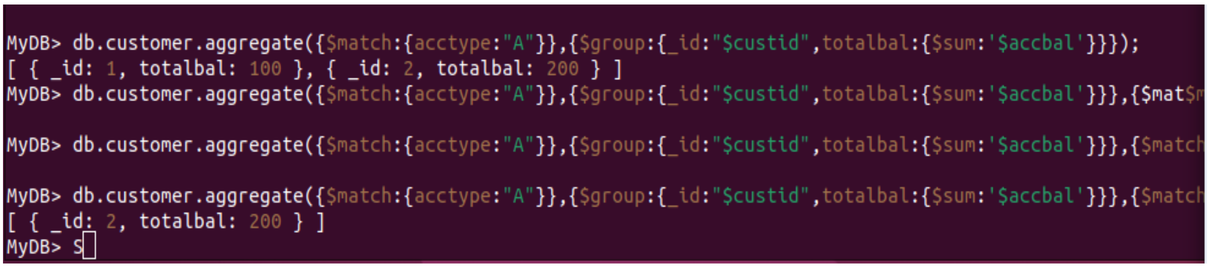
```
[
  { _id: 1, fruits: [ 'grapes', 'banana', 'apple' ] },
  {
    _id: 2,
    fruits: [ 'grapes', 'mango', 'cherry' ],
    price: [ { grapes: 80, mango: 100, cherry: 200 } ]
  },
  { _id: 3, fruits: [ 'banana', 'mango' ] }
]
MyDB> db.food.find().pretty();
[
  { _id: 1, fruits: [ 'grapes', 'banana', 'apple' ] },
  {
    _id: 2,
    fruits: [ 'grapes', 'mango', 'cherry' ],
    price: [ { grapes: 80, mango: 100, cherry: 200 } ]
  },
  { _id: 3, fruits: [ 'banana', 'mango' ] }
]
MyDB> db.createCollection("customer");
{ ok: 1 }
MyDB> var val=[{custid:1,accbal:100,acctype:"A"},{custid:1,accbal:100,acctype:"B"},{custid:2,accbal:200,accty
ype:"B"});
Uncaught:
SyntaxError: Unexpected token, expected "," (1:144)

> 1 | var val=[{custid:1,accbal:100,acctype:"A"},{custid:1,accbal:100,acctype:"B"},{custid:2,accbal:200,accty
ype:"B"});
    |
        ^
  2 |

MyDB> var val=[{custid:1,accbal:100,acctype:"A"},{custid:1,accbal:100,acctype:"B"},{custid:2,accbal:200,accty
ype:"B"}];

MyDB> db.customer.insert(val);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("683405cb126af1945c475870"),
    '1': ObjectId("683405cb126af1945c475871"),
    '2': ObjectId("683405cb126af1945c475872"),
    '3': ObjectId("683405cb126af1945c475873")
  }
}
MyDB> db.customer.aggregate({$group:{_id:'$custid',totalbal:{$sum:'$accbal'}}});
[ { _id: 1, totalbal: 200 }, { _id: 2, totalbal: 400 } ]
MyDB> db.Customers.aggregate ( {$match:{AcctType:”S”}},{$group : { _id : “$custID”,TotAccBal :
Uncaught:
SyntaxError: Unexpected character '”'. (1:43)

> 1 | db.Customers.aggregate ( {$match:{AcctType:”S”}},{$group : { _id : “$custID”,TotAccBal :
```



```
MyDB> db.customer.aggregate({$match:{acctype:"A"}},{$group:{_id:"$custid",totalbal:{$sum:'$accbal'}}});
[ { _id: 1, totalbal: 100 }, { _id: 2, totalbal: 200 } ]
MyDB> db.customer.aggregate({$match:{acctype:"A"}},{$group:{_id:"$custid",totalbal:{$sum:'$accbal'}}},{$mat$m

MyDB> db.customer.aggregate({$match:{acctype:"A"}},{$group:{_id:"$custid",totalbal:{$sum:'$accbal'}}},{$match

MyDB> db.customer.aggregate({$match:{acctype:"A"}},{$group:{_id:"$custid",totalbal:{$sum:'$accbal'}}},{$match
[ { _id: 2, totalbal: 200 } ]
MyDB> S
```