Ex. No.: 6

Import a JASON file from the command line. Apply the following actions with the data present in the JASON file where, projection, aggregation, remove, count, limit, skip and sort

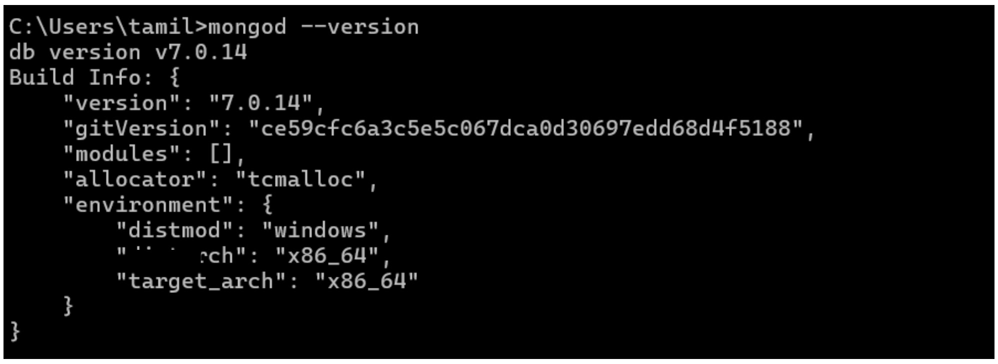
AIM:

To import a JASON file from the command line and apply the following actions with the data present in the JASON file where, projection, aggregation, remove, count, limit, skip and sort.

PROCEDURE:

1. Download and install MongoDB:

1. Visit the MongoDB Download Center and choose the version suitable for your operating system (Windows).
2. Select MSI as the package, and click "Download".
3. Run the downloaded MSI installer. Follow the prompts in the installation wizard.
4. Choose "Complete" for the setup type to install all components.
5. Select the option to install MongoDB as a service (recommended), which allows MongoDB to start automatically with your system.

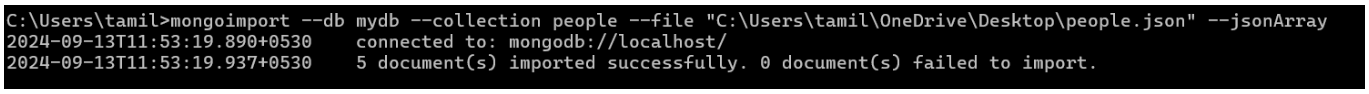


```
C:\Users\tamil>mongod --version
db version v7.0.14
Build Info: {
    "version": "7.0.14",
    "gitVersion": "ce59cfc6a3c5e5c067dca0d30697edd68d4f5188",
    "modules": [],
    "allocator": "tcmalloc",
    "environment": {
        "distmod": "windows",
        "[illegible]ch": "x86_64",
        "target_arch": "x86_64"
    }
}
```

6. Check the version installed by using the command *$ mongod –version*

2. Create a sample people.json file with the following content:

```
[{"name": "Alice",
   "age": 30,
   "city": "New York"},
  {"name": "Bob",
   "age": 25,
   "city": "Los Angeles" },
  {"name": "Charlie",
   "age": 35,
   "city": "Chicago"},
  {"name": "David",
   "age": 28,
   "city": "Houston"},
  {"name": "Eve",
   "age": 22,
```



```
C:\Users\tamil>mongoimport --db mydb --collection people --file "C:\Users\tamil\OneDrive\Desktop\people.json" --jsonArray
2024-09-13T11:53:19.890+0530    connected to: mongodb://localhost/
2024-09-13T11:53:19.937+0530    5 document(s) imported successfully. 0 document(s) failed to import.
```

```
 "city": "Phoenix"}]
```

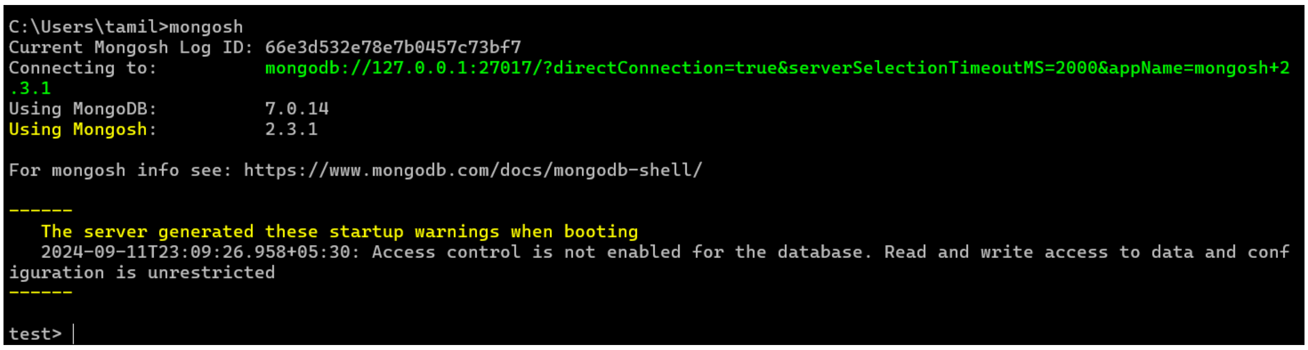
Import the JSON File

To import the JSON file into your MongoDB database, open your terminal or command prompt and use the mongoimport command:

>mongoimport --db mydb --collection people --file path_to_your_json/people.json –jsonArray

3. Download and install MongoShell:

1. Visit the MongoDB Shell https://www.mongodb.com/try/download/shell Download Page.
2. Select your operating system as Windows and download the MSI package. Run the downloaded MSI installer.
3. Follow the installation prompts to complete the installation.
4. Ensure the option to add MongoDB Shell to your PATH is checked during installation.
5. To verify, open Command Prompt or PowerShell and type the following command to check if



```
C:\Users\tamil>mongosh
Current Mongosh Log ID: 66e3d532e78e7b0457c73bf7
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2
.3.1
Using MongoDB:          7.0.14
Using Mongosh:          2.3.1

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

------
   The server generated these startup warnings when booting
   2024-09-11T23:09:26.958+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
------

test>
```

   mongosh is installed correctly:

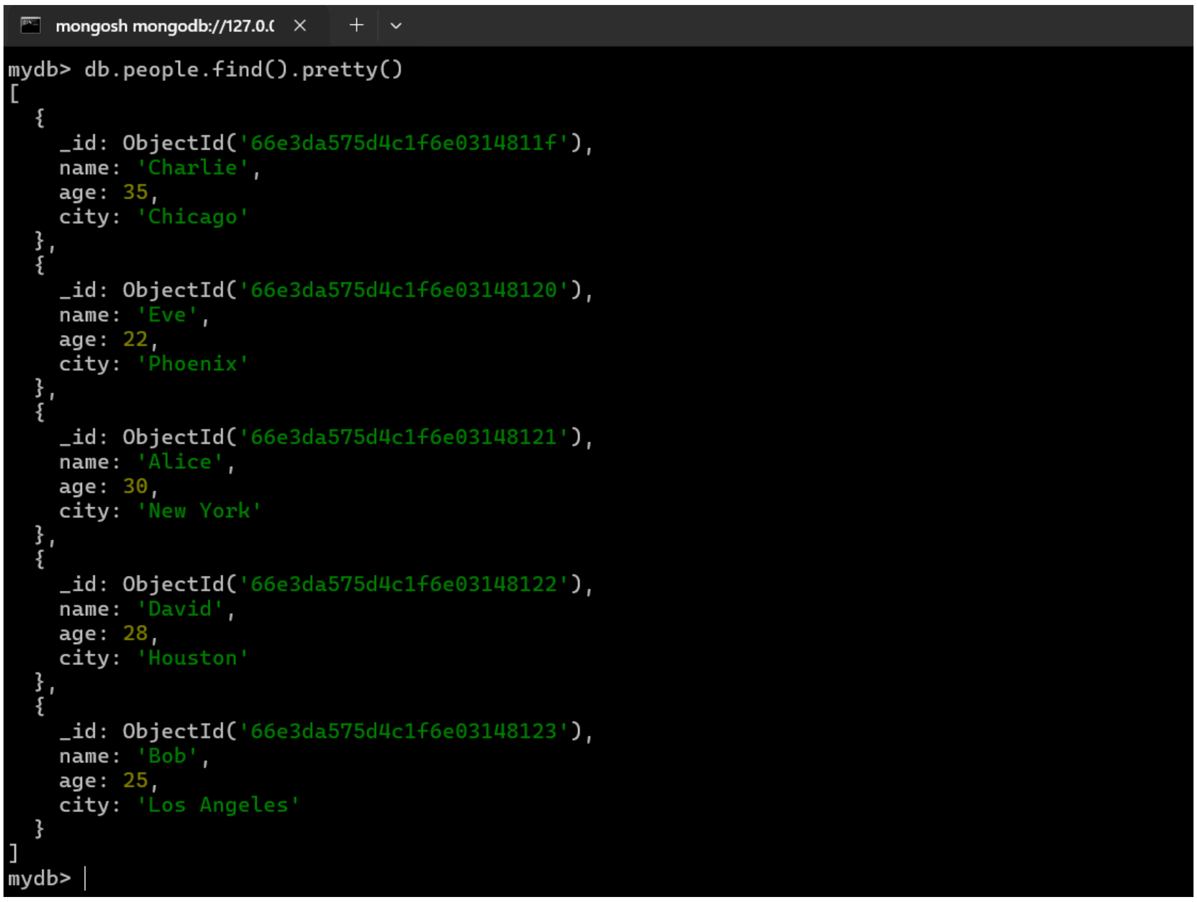
   $ mongosh --version

6. If installed correctly, it will display the version of mongosh.
7. To start MongoDB Shell open Command Prompt Type mongosh to start the MongoDB Shell. It will connect to the default MongoDB server (localhost:27017).

4. Create and Switch to the Database:

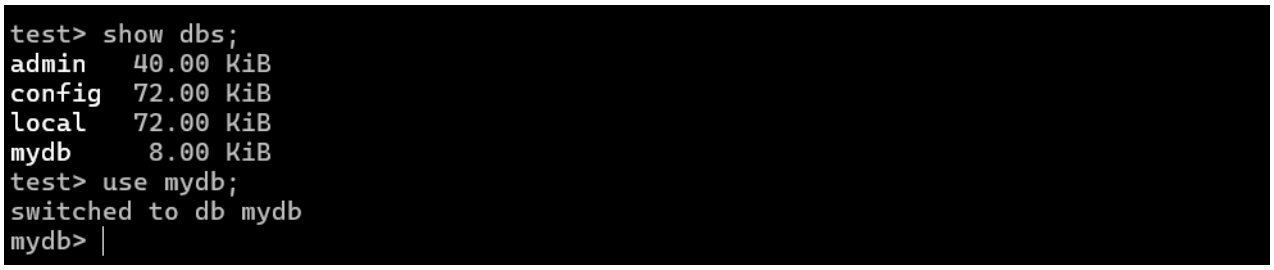
Use the use command to create and switch to the new database. Then, switch to your database and check the collection:

test> use mydb
mydb> show collections

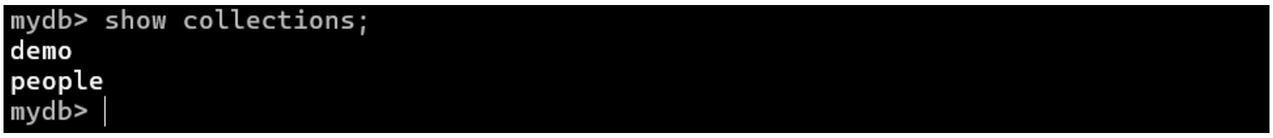


```
mongosh mongodb://127.0.(
mydb> db.people.find().pretty()
[
  {
    _id: ObjectId('66e3da575d4c1f6e0314811f'),
    name: 'Charlie',
    age: 35,
    city: 'Chicago'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148120'),
    name: 'Eve',
    age: 22,
    city: 'Phoenix'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148121'),
    name: 'Alice',
    age: 30,
    city: 'New York'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148122'),
    name: 'David',
    age: 28,
    city: 'Houston'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148123'),
    name: 'Bob',
    age: 25,
    city: 'Los Angeles'
  }
]
mydb>
```

>db.people.find().pretty()

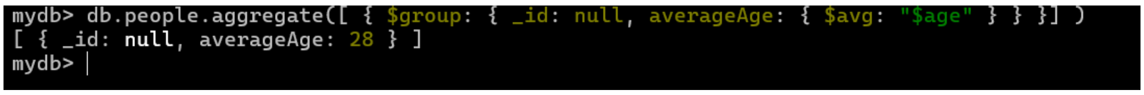


```
test> show dbs;
admin    40.00 KiB
config   72.00 KiB
local    72.00 KiB
mydb      8.00 KiB
test> use mydb;
switched to db mydb
mydb>
```



```
mydb> show collections;
demo
people
mydb>
```

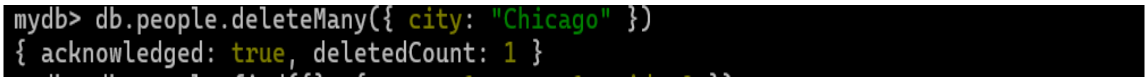
Step 4:



```
mydb> db.people.aggregate([ { $group: { _id: null, averageAge: { $avg: "$age" } } }] )
[ { _id: null, averageAge: 28 } ]
mydb> |
```

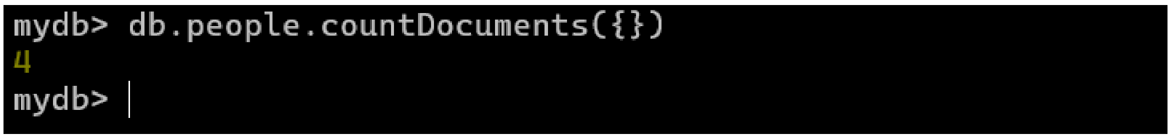
Perform Actions on the JSON Data

Now, we will apply the following operations: where, projection, aggregation, remove, count, limit, skip, and sort.



```
mydb> db.people.deleteMany({ city: "Chicago" })
{ acknowledged: true, deletedCount: 1 }
```

1. **Where (Filter):** Find records where the city is "New York":
   >db.people.find({ city: "New York" })



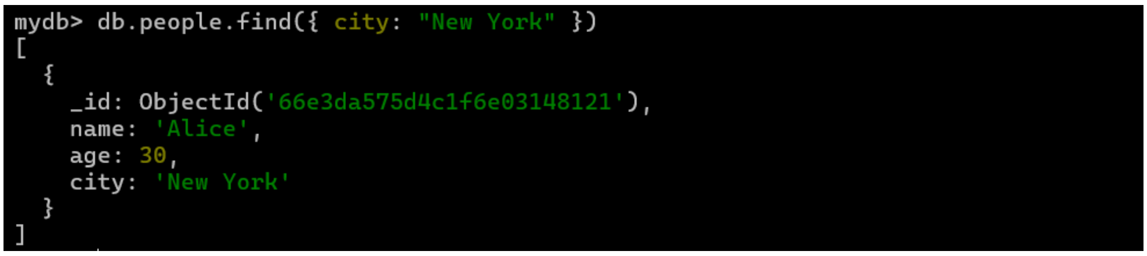
```
mydb> db.people.countDocuments({})
4
mydb> |
```

2. **Projection:** Select only the name and age fields from the records:
   >db.people.find({}, { name: 1, age: 1, _id: 0 })

3. **Aggregation:** Calculate the average age of all people in the collection:
   >db.people.aggregate([
   { $group: { _id: null, averageAge: { $avg: "$age" } } }
   ])

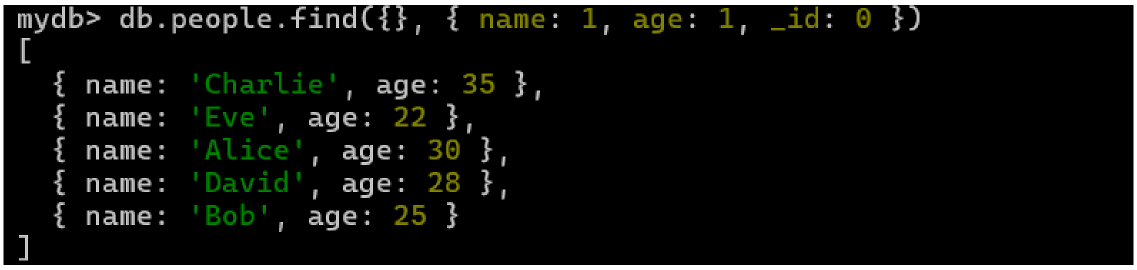
4. **Remove:** Delete people who live in "Chicago":
   >db.people.deleteMany({ city: "Chicago" })

5. **Count:** Count the total number of people in the collection:
   >db.people.countDocuments({})



```
mydb> db.people.find({ city: "New York" })
[
  {
    _id: ObjectId('66e3da575d4c1f6e03148121'),
    name: 'Alice',
    age: 30,
    city: 'New York'
  }
]
```

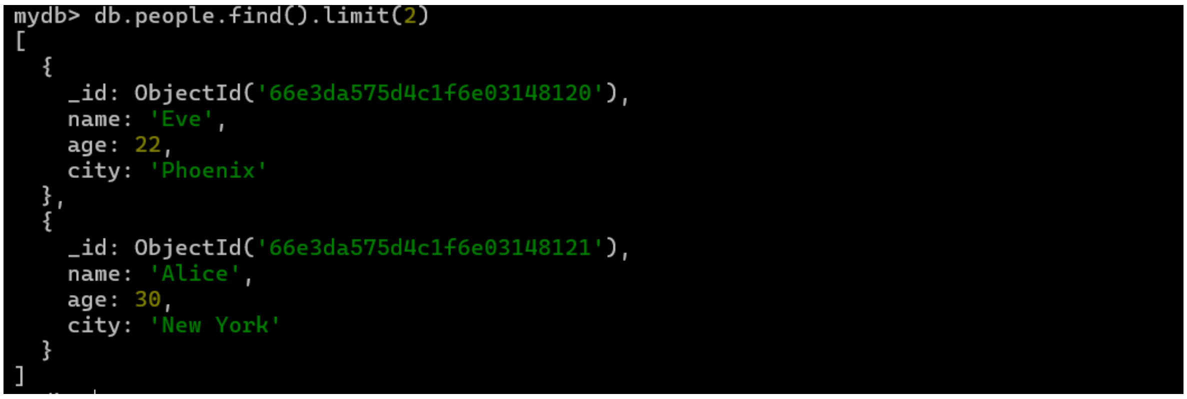
6. **Limit:** Limit the results to 2 records:



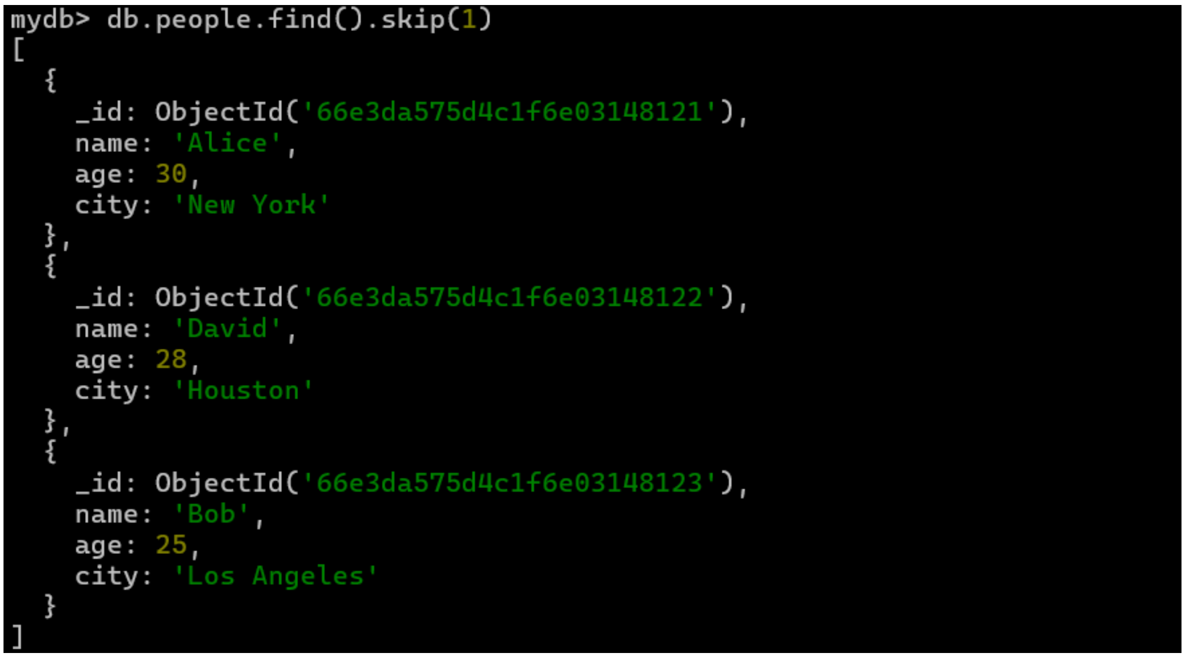
```
mydb> db.people.find({}, { name: 1, age: 1, _id: 0 })
[
  { name: 'Charlie', age: 35 },
  { name: 'Eve', age: 22 },
  { name: 'Alice', age: 30 },
  { name: 'David', age: 28 },
  { name: 'Bob', age: 25 }
]
```

>db.people.find().limit(2)

7. **Skip:** Skip the first record and show the rest:
   >db.people.find().skip(1)
8. **Sort:** Sort the records by age in ascending order:
   >db.people.find().sort({ age: 1 })



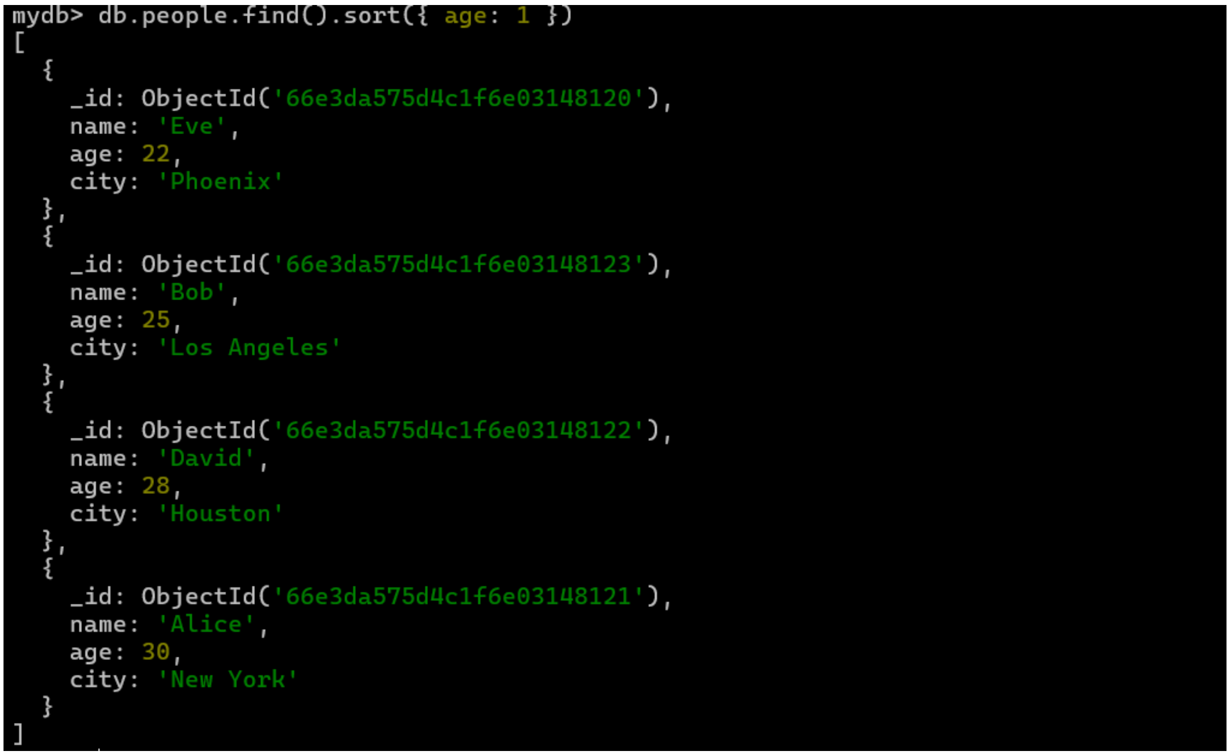
```
mydb> db.people.find().limit(2)
[
  {
    _id: ObjectId('66e3da575d4c1f6e03148120'),
    name: 'Eve',
    age: 22,
    city: 'Phoenix'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148121'),
    name: 'Alice',
    age: 30,
    city: 'New York'
  }
]
```



```
mydb> db.people.find().skip(1)
[
  {
    _id: ObjectId('66e3da575d4c1f6e03148121'),
    name: 'Alice',
    age: 30,
    city: 'New York'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148122'),
    name: 'David',
    age: 28,
    city: 'Houston'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148123'),
    name: 'Bob',
    age: 25,
    city: 'Los Angeles'
  }
]
```

RESULT:

Thus to import a JASON file from the command line and apply the following actions with the data



```
mydb> db.people.find().sort({ age: 1 })
[
  {
    _id: ObjectId('66e3da575d4c1f6e03148120'),
    name: 'Eve',
    age: 22,
    city: 'Phoenix'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148123'),
    name: 'Bob',
    age: 25,
    city: 'Los Angeles'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148122'),
    name: 'David',
    age: 28,
    city: 'Houston'
  },
  {
    _id: ObjectId('66e3da575d4c1f6e03148121'),
    name: 'Alice',
    age: 30,
    city: 'New York'
  }
]
```

present in the JASON file where, projection, aggregation, remove, count, limit, skip and sort has been executed and verified successfully.