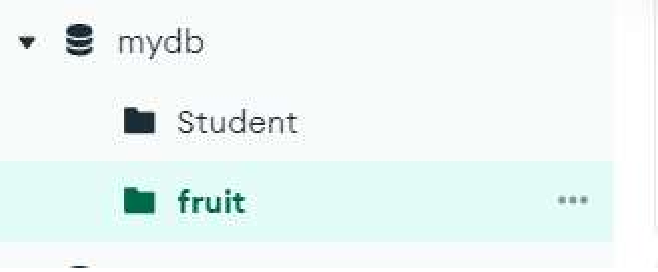
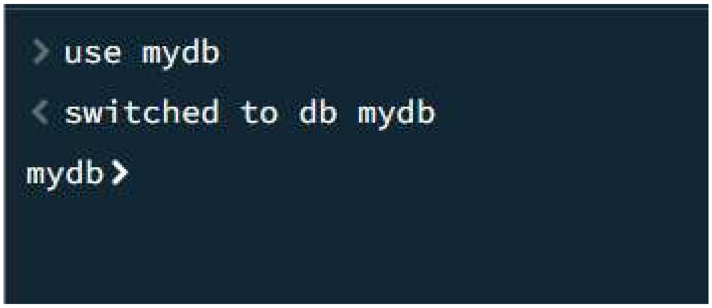
procedure:

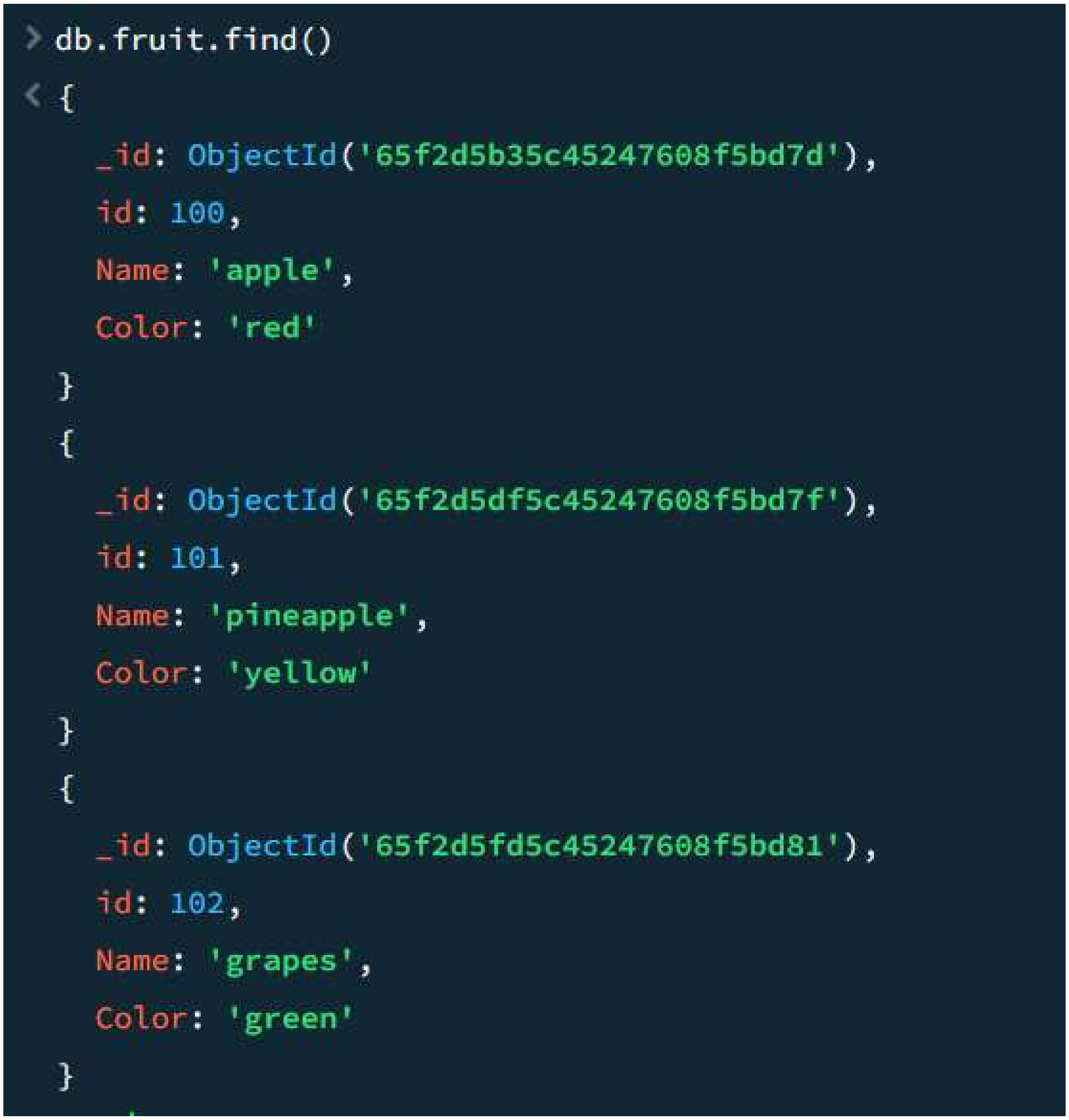
1.use command





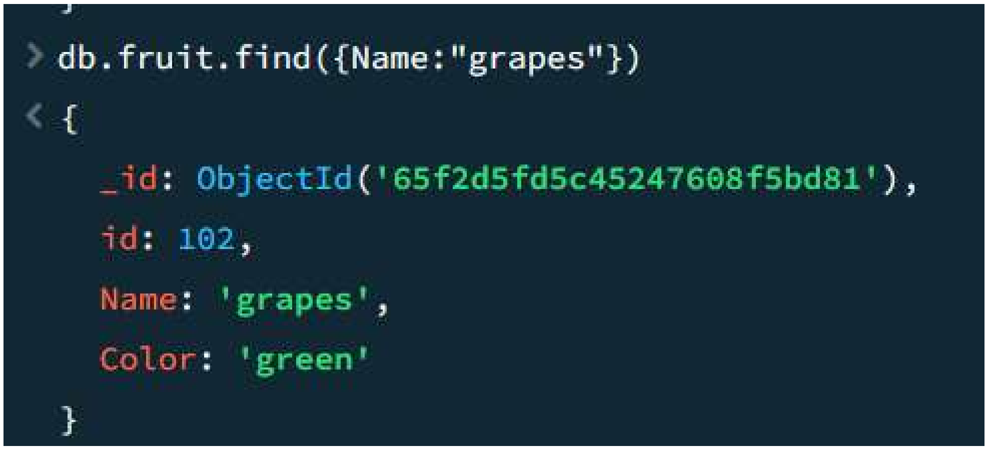
```
> use mydb
< switched to db mydb
mydb>
```

2.find command



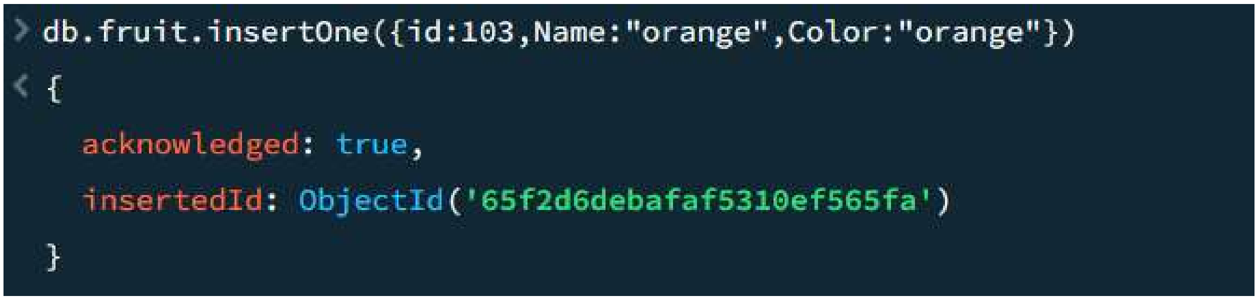
```
> db.fruit.find()
< {
    _id: ObjectId('65f2d5b35c45247608f5bd7d'),
    id: 100,
    Name: 'apple',
    Color: 'red'
  }
  {
    _id: ObjectId('65f2d5df5c45247608f5bd7f'),
    id: 101,
    Name: 'pineapple',
    Color: 'yellow'
  }
  {
    _id: ObjectId('65f2d5fd5c45247608f5bd81'),
    id: 102,
    Name: 'grapes',
    Color: 'green'
  }
```

3.find using condition

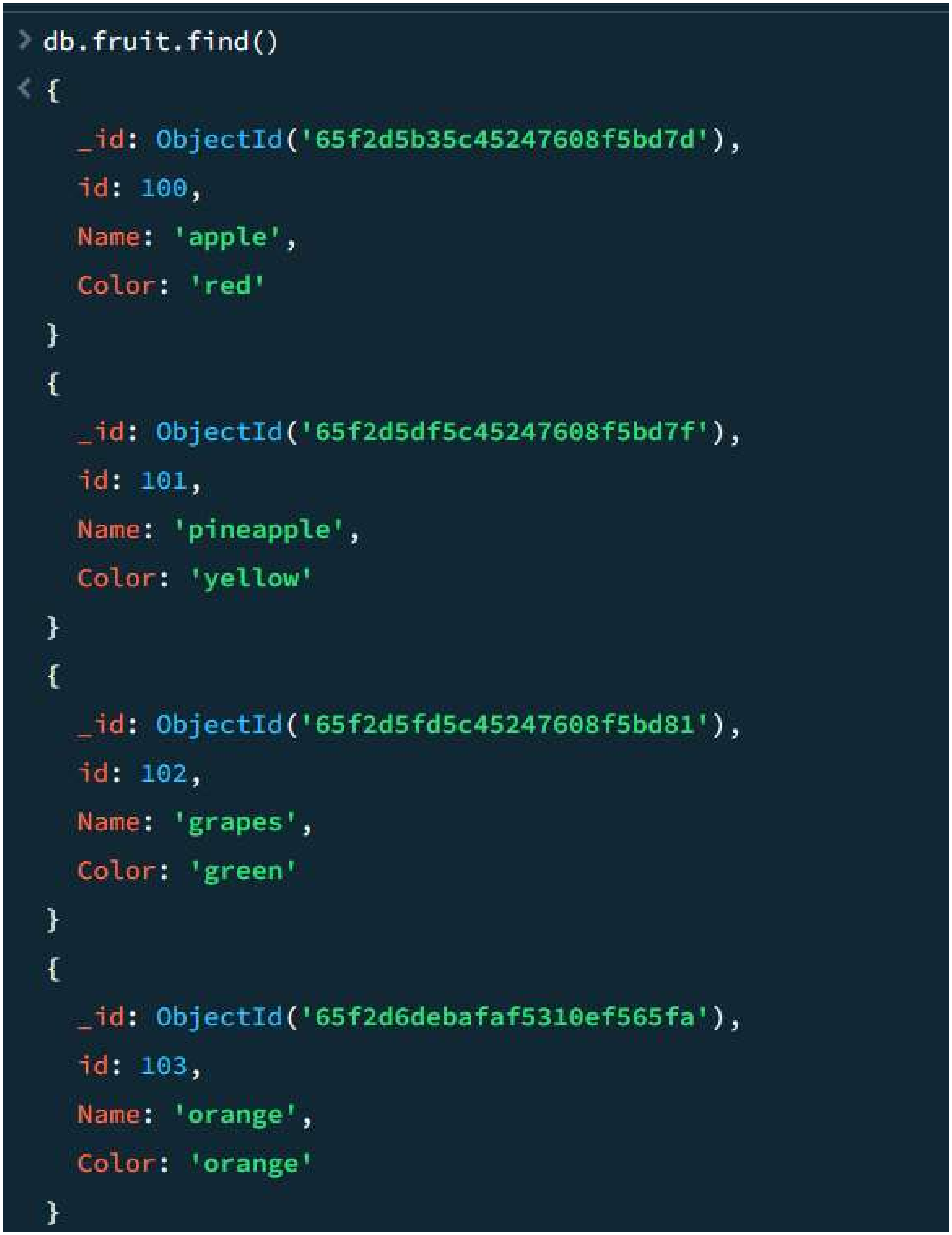


```
> db.fruit.find({Name:"grapes"})
< {
    _id: ObjectId('65f2d5fd5c45247608f5bd81'),
    id: 102,
    Name: 'grapes',
    Color: 'green'
  }
```

4.inserting one document

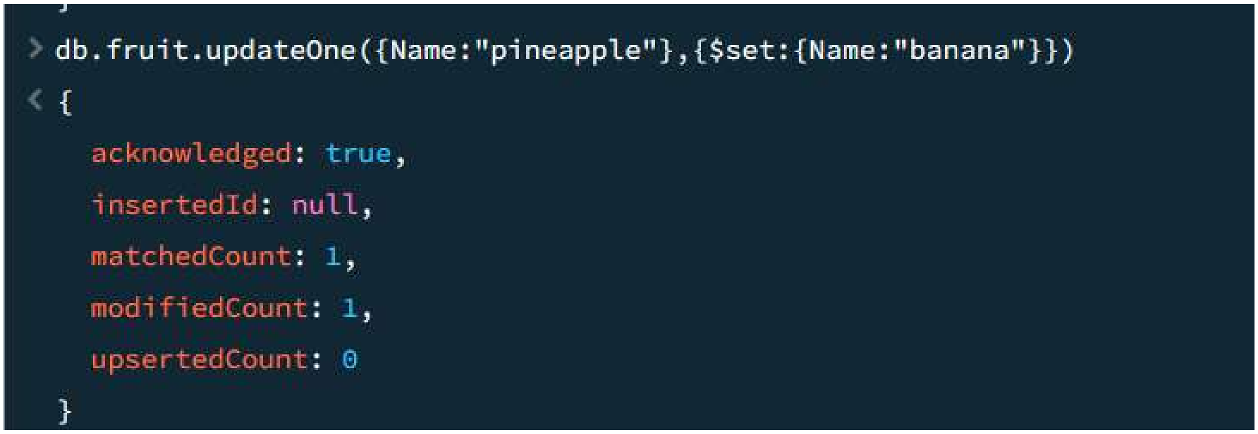


```
> db.fruit.insertOne({id:103,Name:"orange",Color:"orange"})
< {
    acknowledged: true,
    insertedId: ObjectId('65f2d6debafaf5310ef565fa')
  }
```

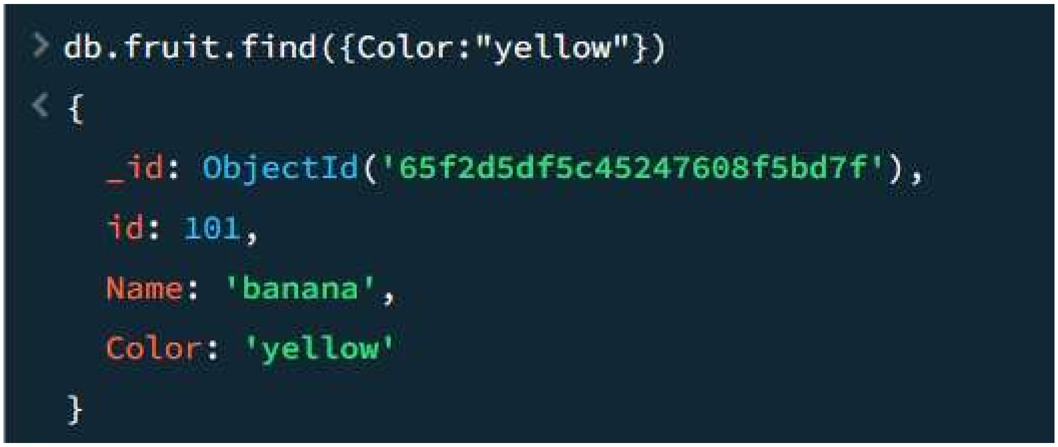


```
> db.fruit.find()
< {
    _id: ObjectId('65f2d5b35c45247608f5bd7d'),
    id: 100,
    Name: 'apple',
    Color: 'red'
  }
  {
    _id: ObjectId('65f2d5df5c45247608f5bd7f'),
    id: 101,
    Name: 'pineapple',
    Color: 'yellow'
  }
  {
    _id: ObjectId('65f2d5fd5c45247608f5bd81'),
    id: 102,
    Name: 'grapes',
    Color: 'green'
  }
  {
    _id: ObjectId('65f2d6debafaf5310ef565fa'),
    id: 103,
    Name: 'orange',
    Color: 'orange'
  }
```

5.updating one document

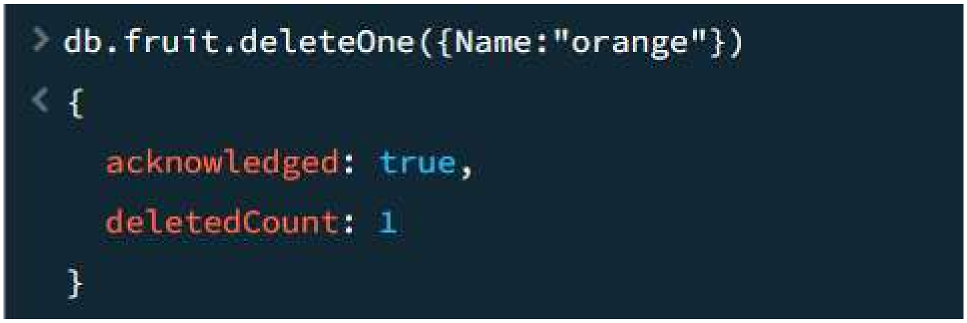


```
> db.fruit.updateOne({Name:"pineapple"},{$set:{Name:"banana"}})
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```



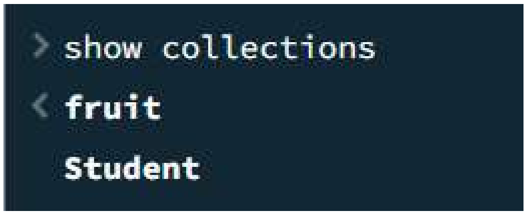
```
> db.fruit.find({Color:"yellow"})
< {
    _id: ObjectId('65f2d5df5c45247608f5bd7f'),
    id: 101,
    Name: 'banana',
    Color: 'yellow'
  }
```

6.delete one document



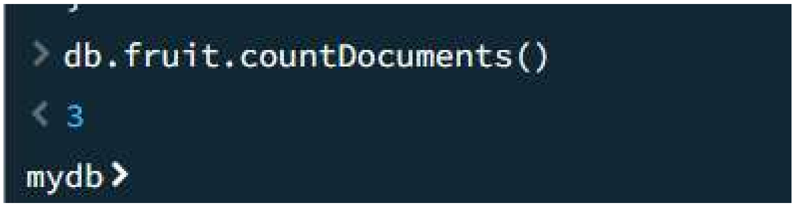
```
> db.fruit.deleteOne({Name:"orange"})
< {
    acknowledged: true,
    deletedCount: 1
  }
```

7.show collections



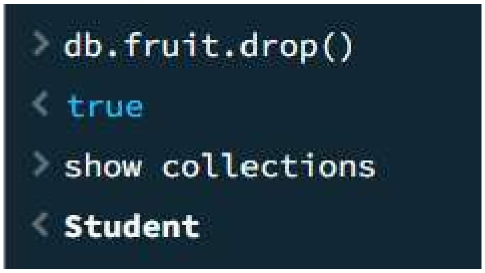
```
> show collections
< fruit
  Student
```

8.countDocuments



```
  }
> db.fruit.countDocuments()
< 3
mydb>
```

9.drop collection



```
> db.fruit.drop()
< true
> show collections
< Student
```

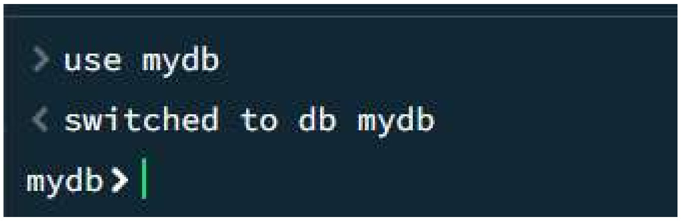
Experiment 27

19-03-24

Aim: Build sample collection of documents using shell commands in mongosh and perform indexing

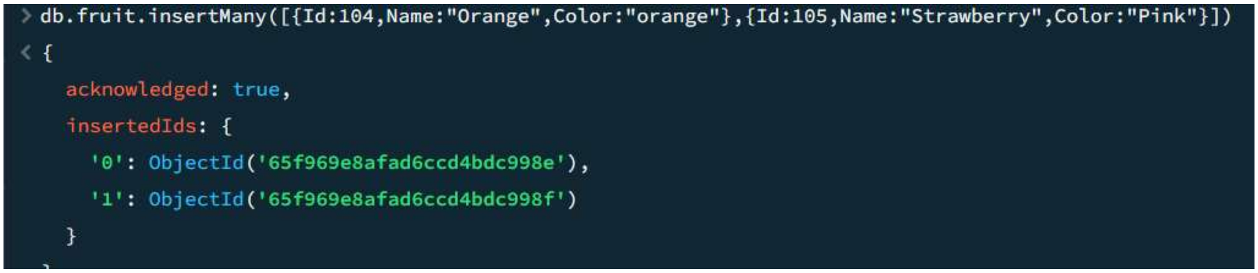
CO4

(i) To switch to sampledb :



```
> use mydb
< switched to db mydb
mydb>
```

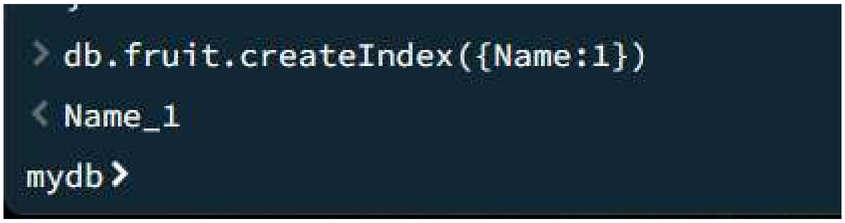
(i) To insert documents



```
> db.fruit.insertMany([{Id:104,Name:"Orange",Color:"orange"},{Id:105,Name:"Strawberry",Color:"Pink"}])
< {
    acknowledged: true,
    insertedIds: {
      '0': ObjectId('65f969e8afad6ccd4bdc998e'),
      '1': ObjectId('65f969e8afad6ccd4bdc998f')
    }
```

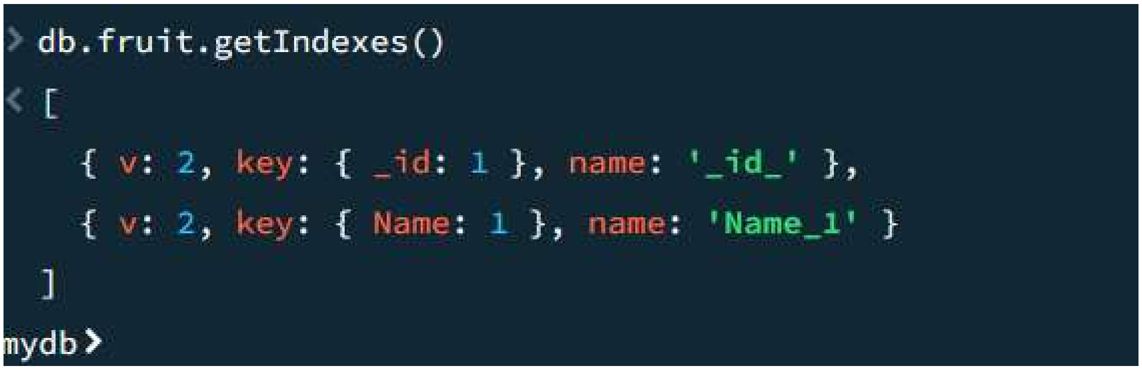
(iii)Perform Indexing Operations:

Create Index:



```
> db.fruit.createIndex({Name:1})
< Name_1
mydb>
```

Get Index



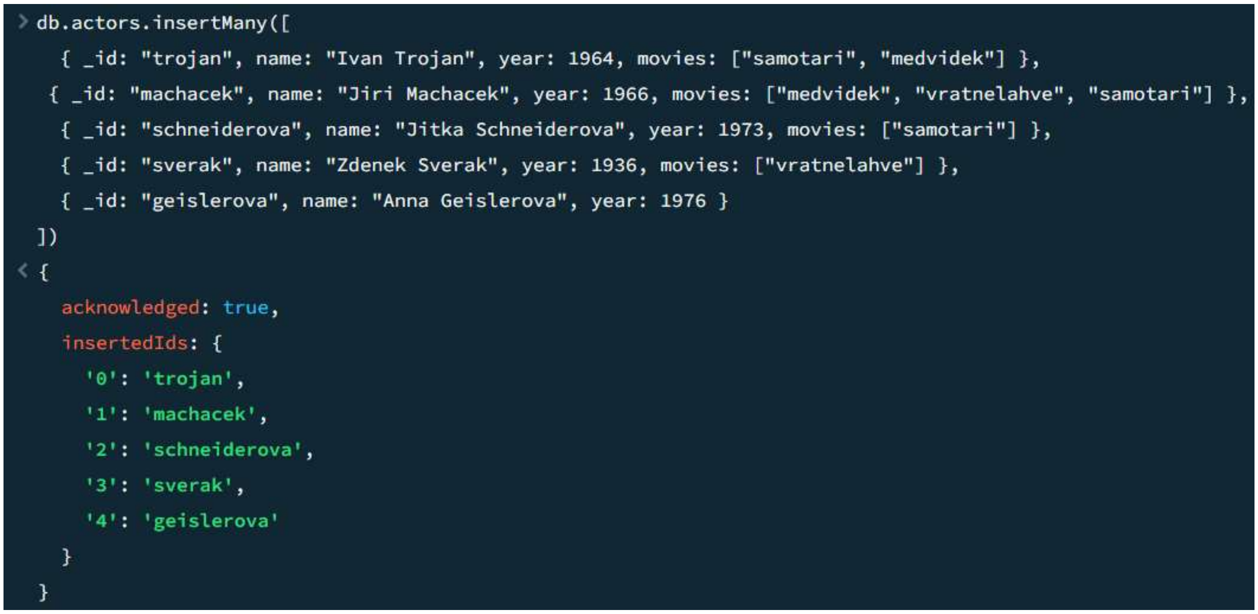
```
> db.fruit.getIndexes()
< [
    { v: 2, key: { _id: 1 }, name: '_id_' },
    { v: 2, key: { Name: 1 }, name: 'Name_1' }
  ]
mydb>
```

Experiment 28

Aim: Buiild sample collection of documents and perform various shell commands in mongosh

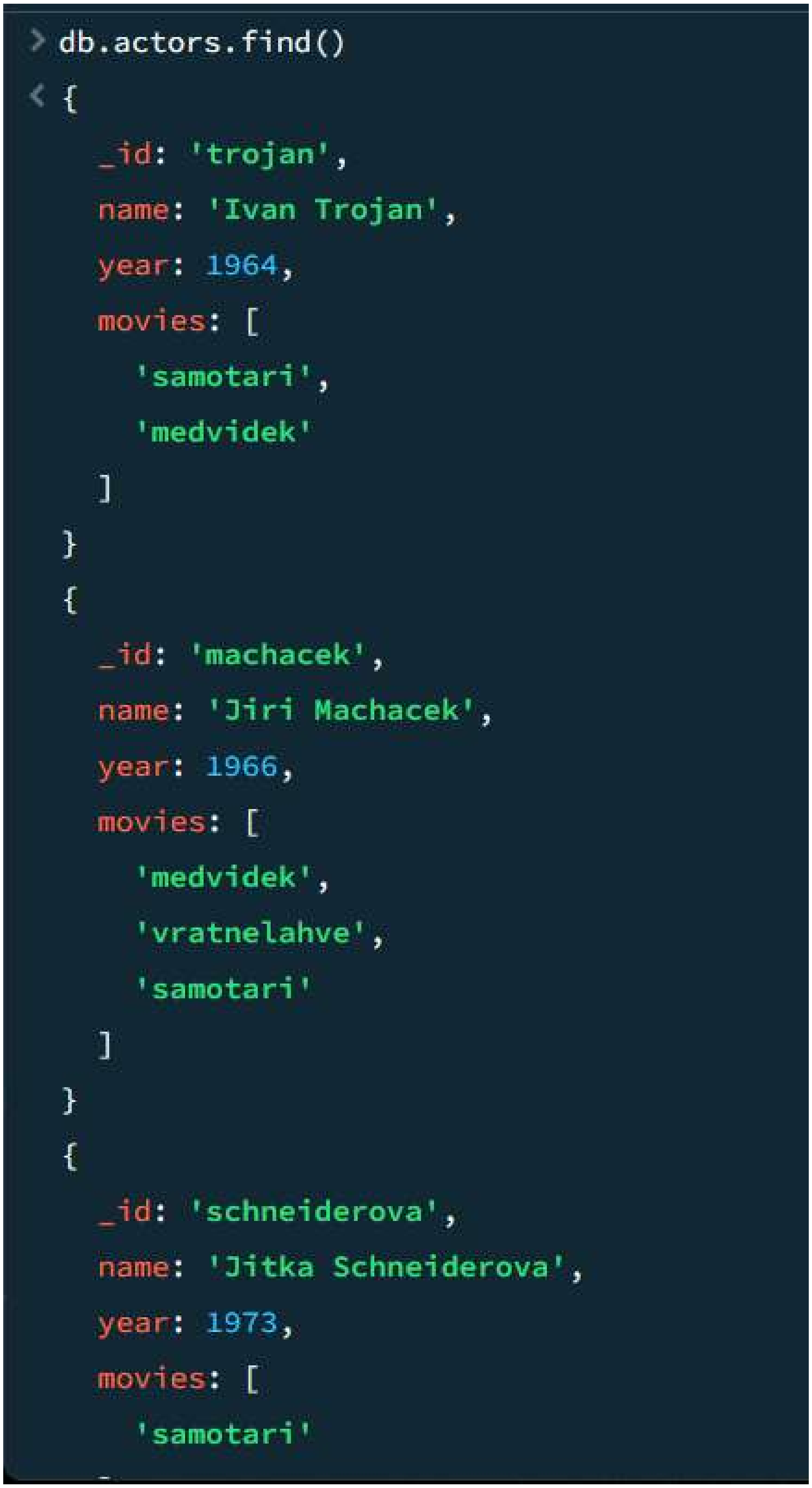
CO4

(i) Insert multiple documents

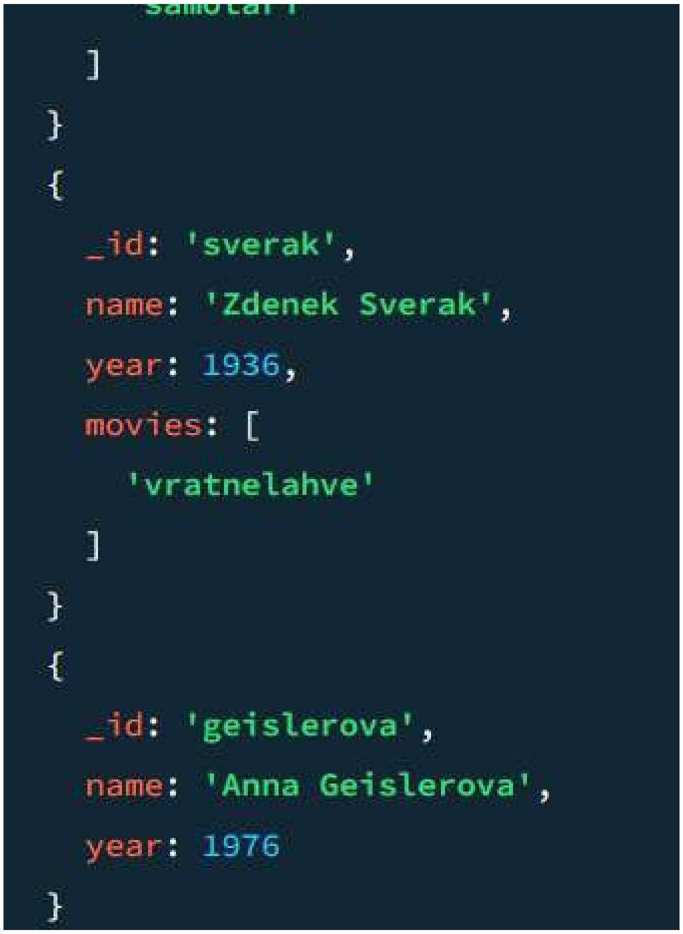


```
> db.actors.insertMany([
    { _id: "trojan", name: "Ivan Trojan", year: 1964, movies: ["samotari", "medvidek"] },
   { _id: "machacek", name: "Jiri Machacek", year: 1966, movies: ["medvidek", "vratnelahve", "samotari"] },
    { _id: "schneiderova", name: "Jitka Schneiderova", year: 1973, movies: ["samotari"] },
    { _id: "sverak", name: "Zdenek Sverak", year: 1936, movies: ["vratnelahve"] },
    { _id: "geislerova", name: "Anna Geislerova", year: 1976 }
  ])
< {
    acknowledged: true,
    insertedIds: {
      '0': 'trojan',
      '1': 'machacek',
      '2': 'schneiderova',
      '3': 'sverak',
      '4': 'geislerova'
    }
  }
```

(ii)To return all documents from the a collection:

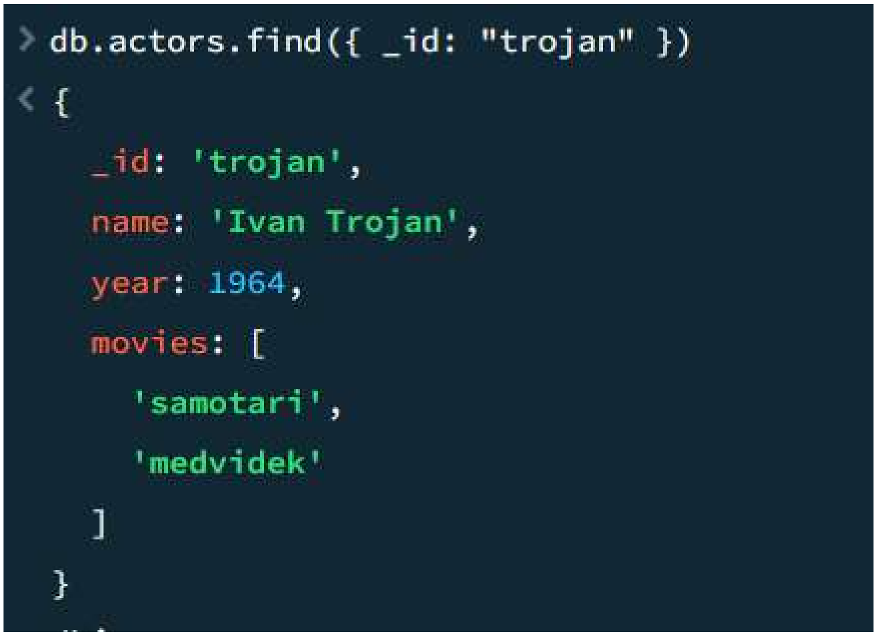


```
> db.actors.find()
< {
    _id: 'trojan',
    name: 'Ivan Trojan',
    year: 1964,
    movies: [
      'samotari',
      'medvidek'
    ]
  }
  {
    _id: 'machacek',
    name: 'Jiri Machacek',
    year: 1966,
    movies: [
      'medvidek',
      'vratnelahve',
      'samotari'
    ]
  }
  {
    _id: 'schneiderova',
    name: 'Jitka Schneiderova',
    year: 1973,
    movies: [
      'samotari'
```



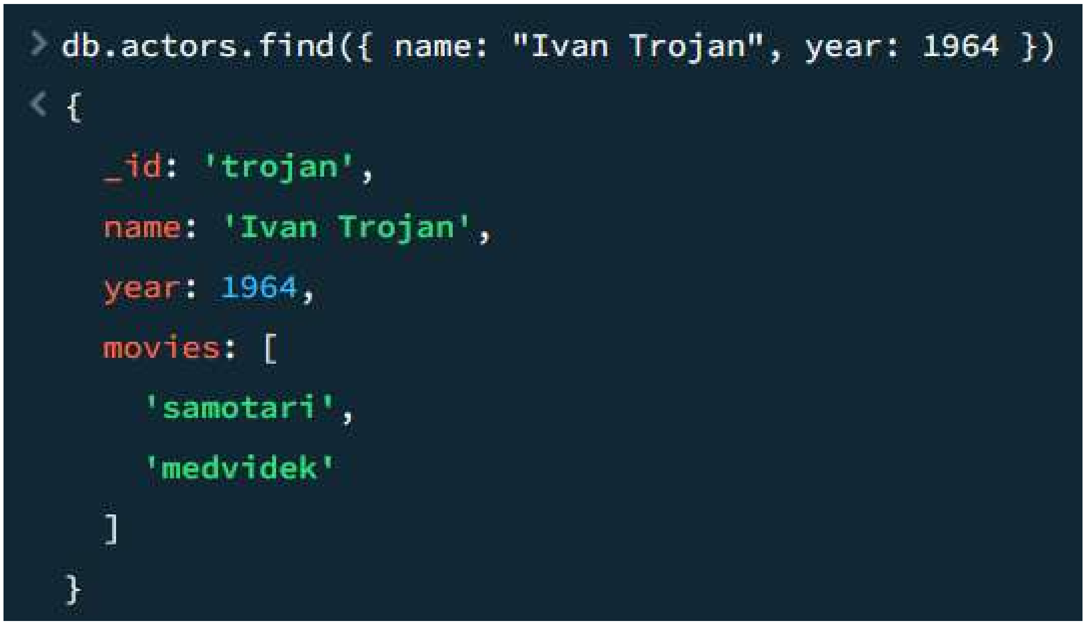
```
      'samotari'
    ]
  }
  {
    _id: 'sverak',
    name: 'Zdenek Sverak',
    year: 1936,
    movies: [
      'vratnelahve'
    ]
  }
  {
    _id: 'geislerova',
    name: 'Anna Geislerova',
    year: 1976
  }
```

(ii) Retrieves documents with the "_id" field equal to "trojan".



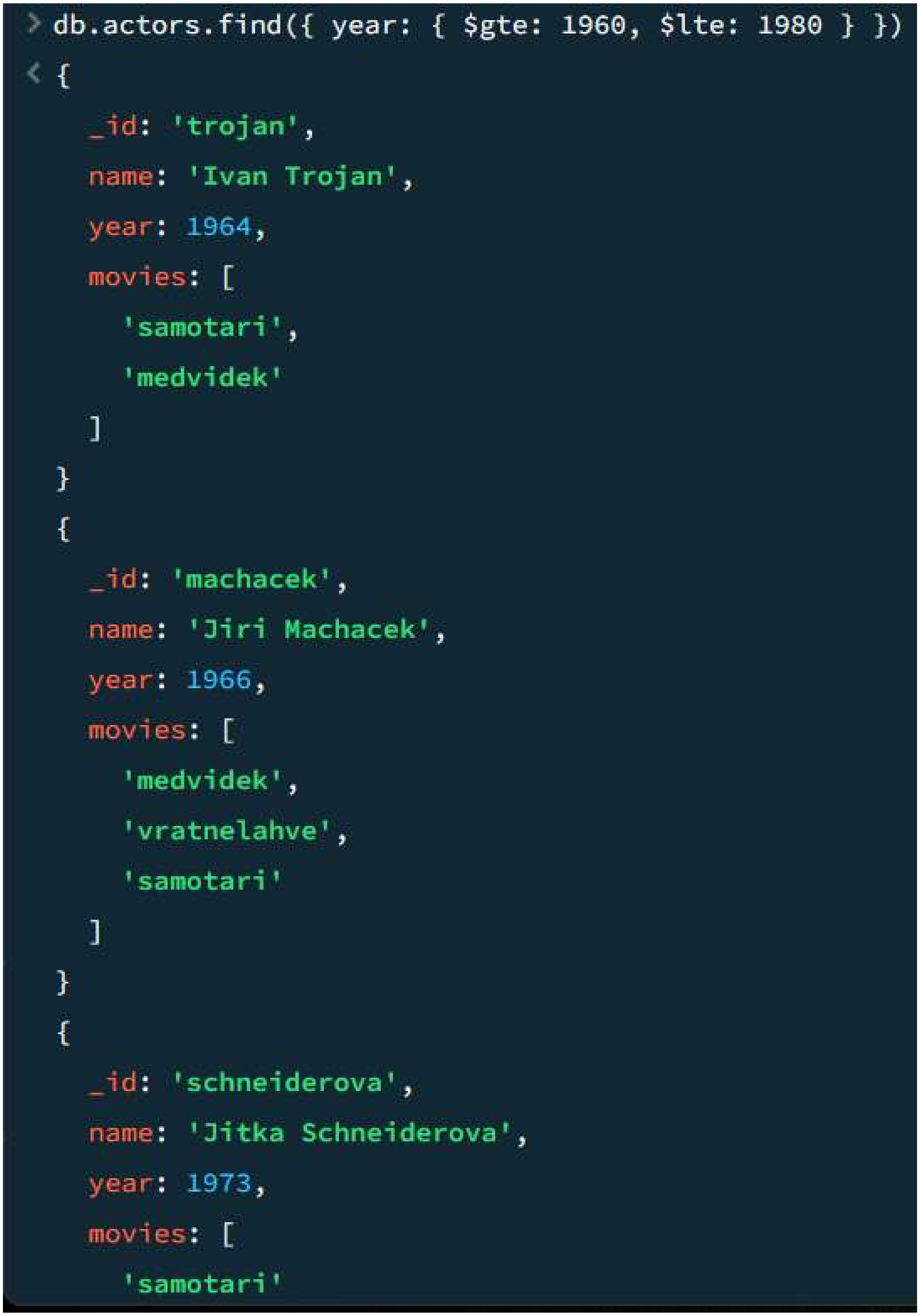
```
> db.actors.find({ _id: "trojan" })
< {
    _id: 'trojan',
    name: 'Ivan Trojan',
    year: 1964,
    movies: [
      'samotari',
      'medvidek'
    ]
  }
```

(iii) Retrieves documents with the "name" field equal to "Ivan Trojan" and the "year" field equal to 1964.

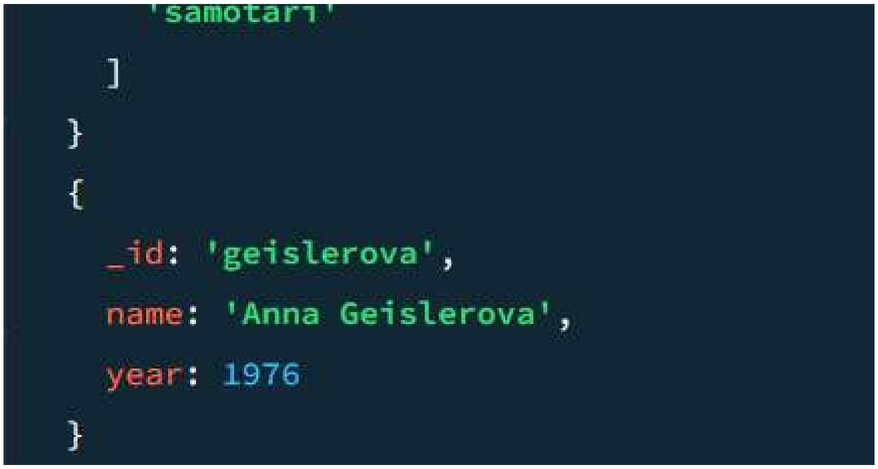


```
> db.actors.find({ name: "Ivan Trojan", year: 1964 })
< {
    _id: 'trojan',
    name: 'Ivan Trojan',
    year: 1964,
    movies: [
      'samotari',
      'medvidek'
    ]
  }
```

(iv) Retrieves documents with the "year" field between 1960 and 1980 (inclusive).

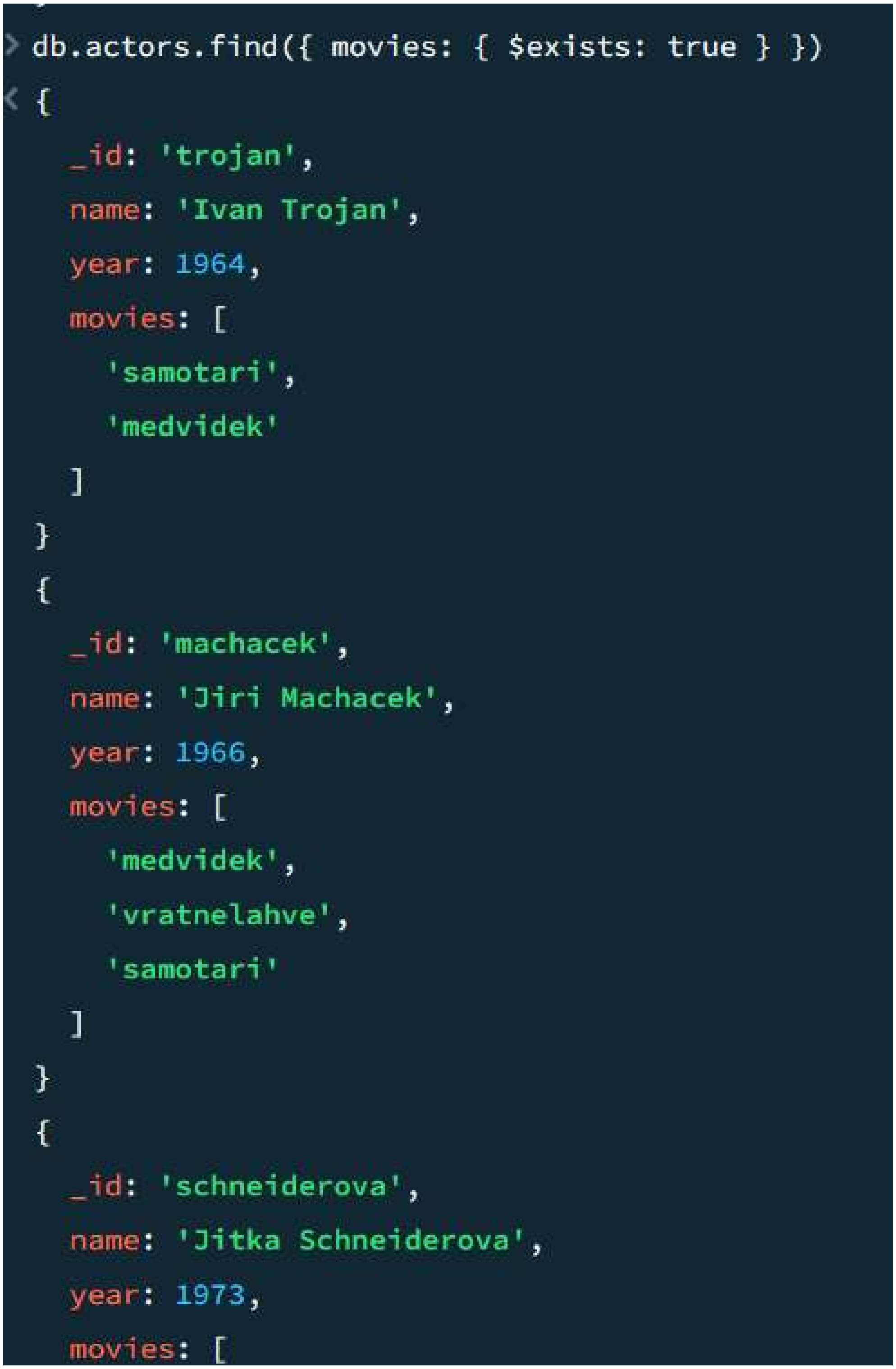


```
> db.actors.find({ year: { $gte: 1960, $lte: 1980 } })
< {
    _id: 'trojan',
    name: 'Ivan Trojan',
    year: 1964,
    movies: [
      'samotari',
      'medvidek'
    ]
  }
  {
    _id: 'machacek',
    name: 'Jiri Machacek',
    year: 1966,
    movies: [
      'medvidek',
      'vratnelahve',
      'samotari'
    ]
  }
  {
    _id: 'schneiderova',
    name: 'Jitka Schneiderova',
    year: 1973,
    movies: [
      'samotari'
```

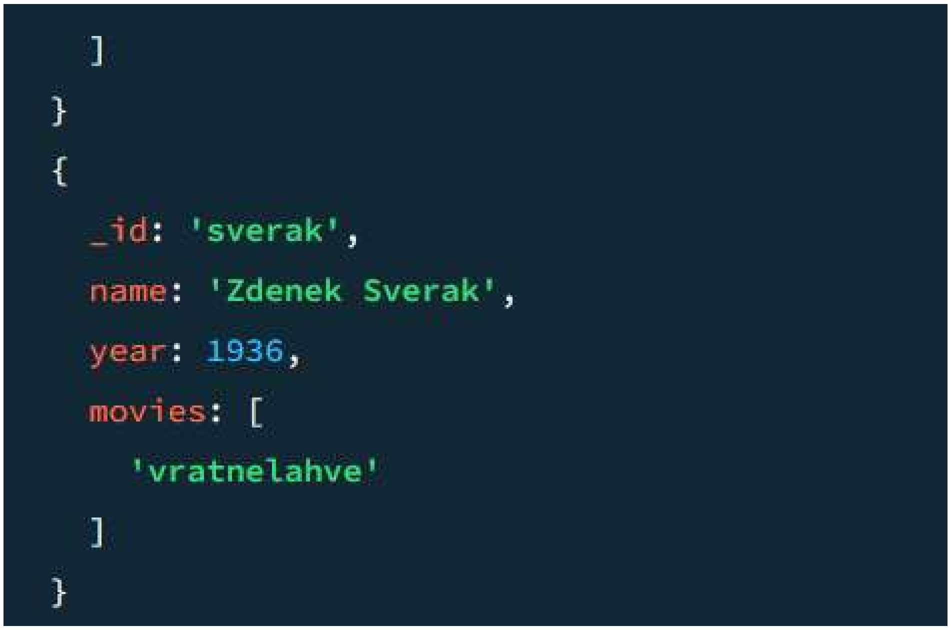


```
      'samotari'
    ]
  }
  {
    _id: 'geislerova',
    name: 'Anna Geislerova',
    year: 1976
  }
```

(v) Retrieves documents where the "movies" field exists.

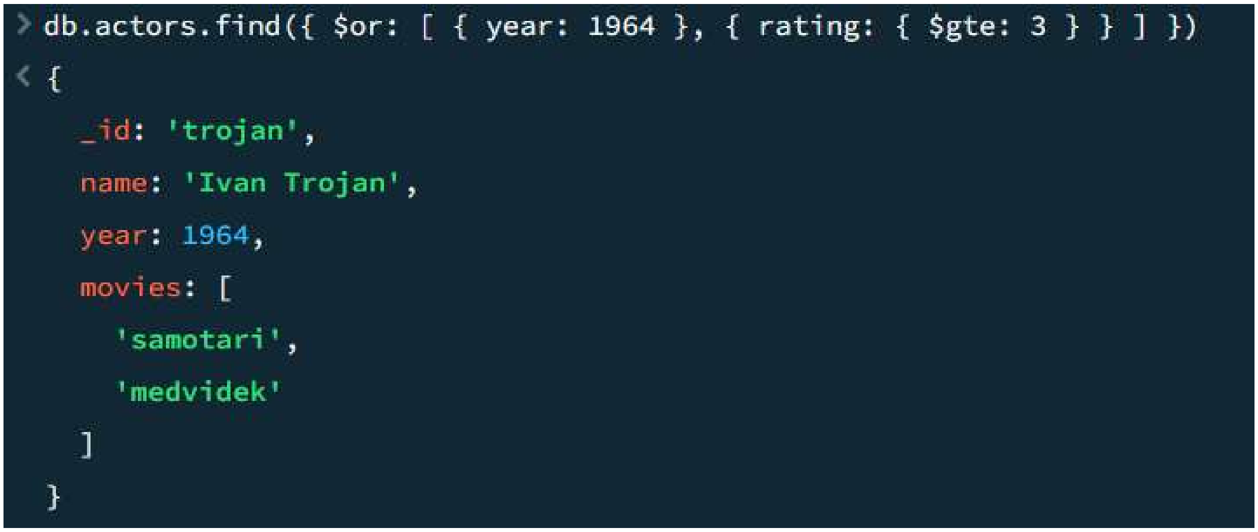


```
> db.actors.find({ movies: { $exists: true } })
< {
    _id: 'trojan',
    name: 'Ivan Trojan',
    year: 1964,
    movies: [
      'samotari',
      'medvidek'
    ]
  }
  {
    _id: 'machacek',
    name: 'Jiri Machacek',
    year: 1966,
    movies: [
      'medvidek',
      'vratnelahve',
      'samotari'
    ]
  }
  {
    _id: 'schneiderova',
    name: 'Jitka Schneiderova',
    year: 1973,
    movies: [
```



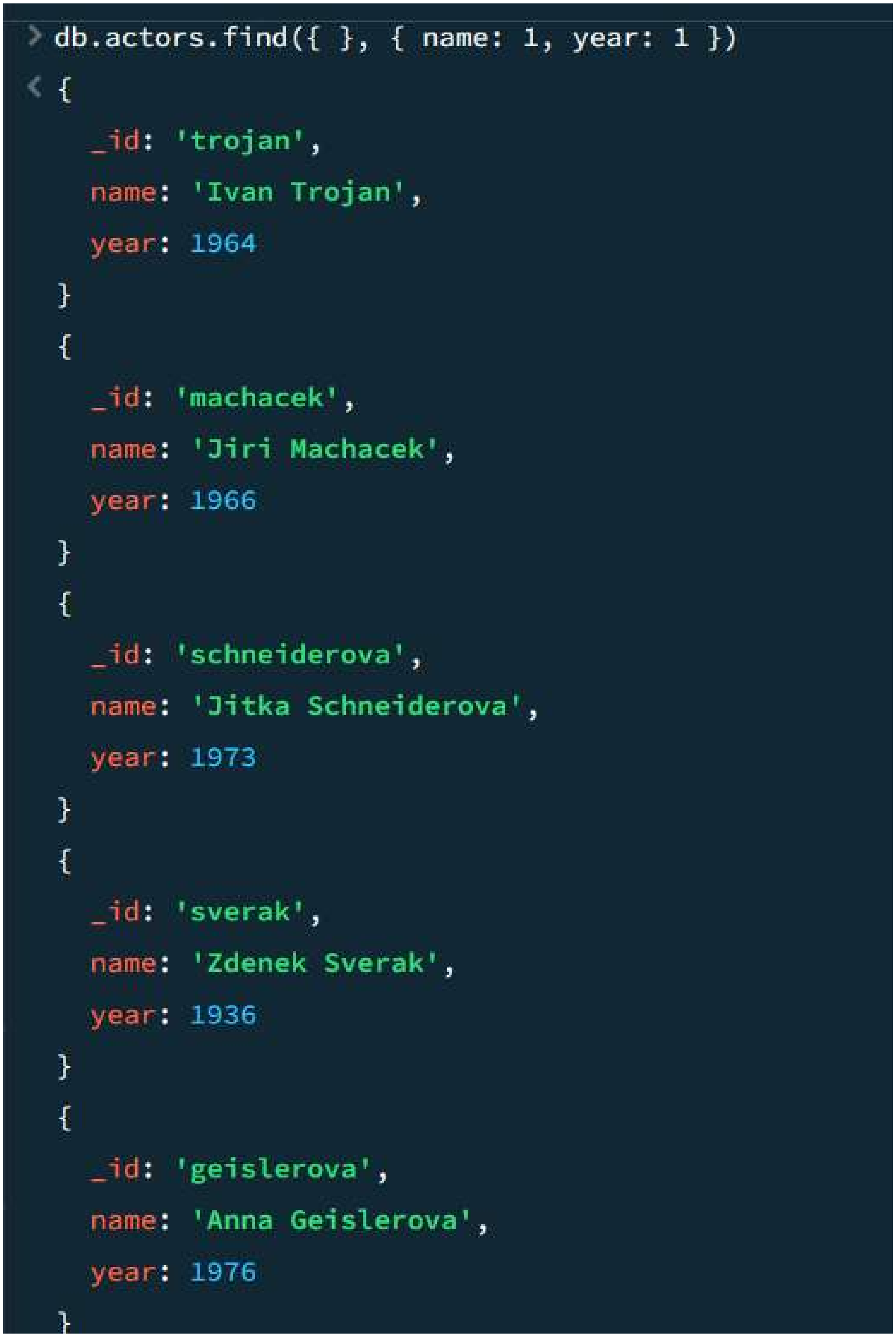
```
    ]
  }
  {
    _id: 'sverak',
    name: 'Zdenek Sverak',
    year: 1936,
    movies: [
      'vratnelahve'
    ]
  }
```

(vi) Retrieves documents where either the "year" field is 1964 or the "rating" field is greater than or equal to 3.



```
> db.actors.find({ $or: [ { year: 1964 }, { rating: { $gte: 3 } } ] })
< {
    _id: 'trojan',
    name: 'Ivan Trojan',
    year: 1964,
    movies: [
      'samotari',
      'medvidek'
    ]
  }
```

(vii) Retrieves all documents and includes only the "name" and "year" fields.



```
> db.actors.find({ }, { name: 1, year: 1 })
< {
    _id: 'trojan',
    name: 'Ivan Trojan',
    year: 1964
  }
  {
    _id: 'machacek',
    name: 'Jiri Machacek',
    year: 1966
  }
  {
    _id: 'schneiderova',
    name: 'Jitka Schneiderova',
    year: 1973
  }
  {
    _id: 'sverak',
    name: 'Zdenek Sverak',
    year: 1936
  }
  {
    _id: 'geislerova',
    name: 'Anna Geislerova',
    year: 1976
  }
```