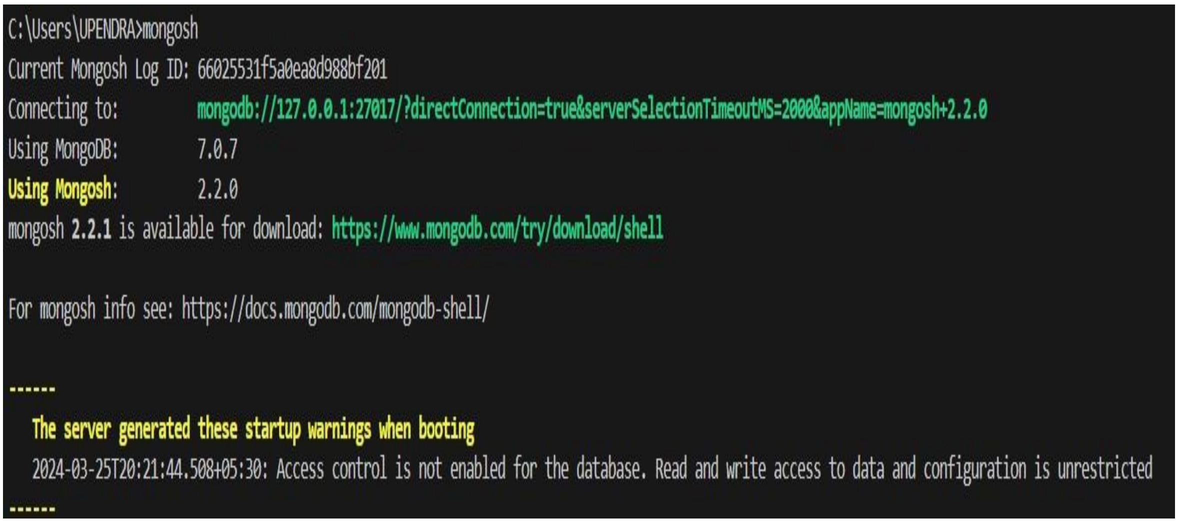
-Assignment: Creating a Database Using MongoDB and Mongosh

MongoDB: MongoDB is a source -available, cross-platform, document-oriented database program. Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with optional schemas.

MongoDB is developed by MongoDB Inc.

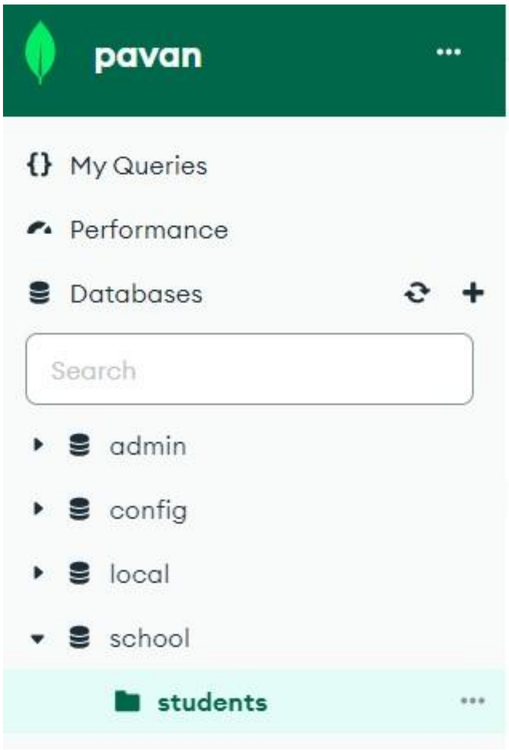


```
C:\Users\UPENDRA>mongosh
Current Mongosh Log ID: 66025531f5a0ea8d988bf201
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.0
Using MongoDB:          7.0.7
Using Mongosh:          2.2.0
mongosh 2.2.1 is available for download: 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
   2024-03-25T20:21:44.508+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------
```

ASSIGNMENT DETAILS:

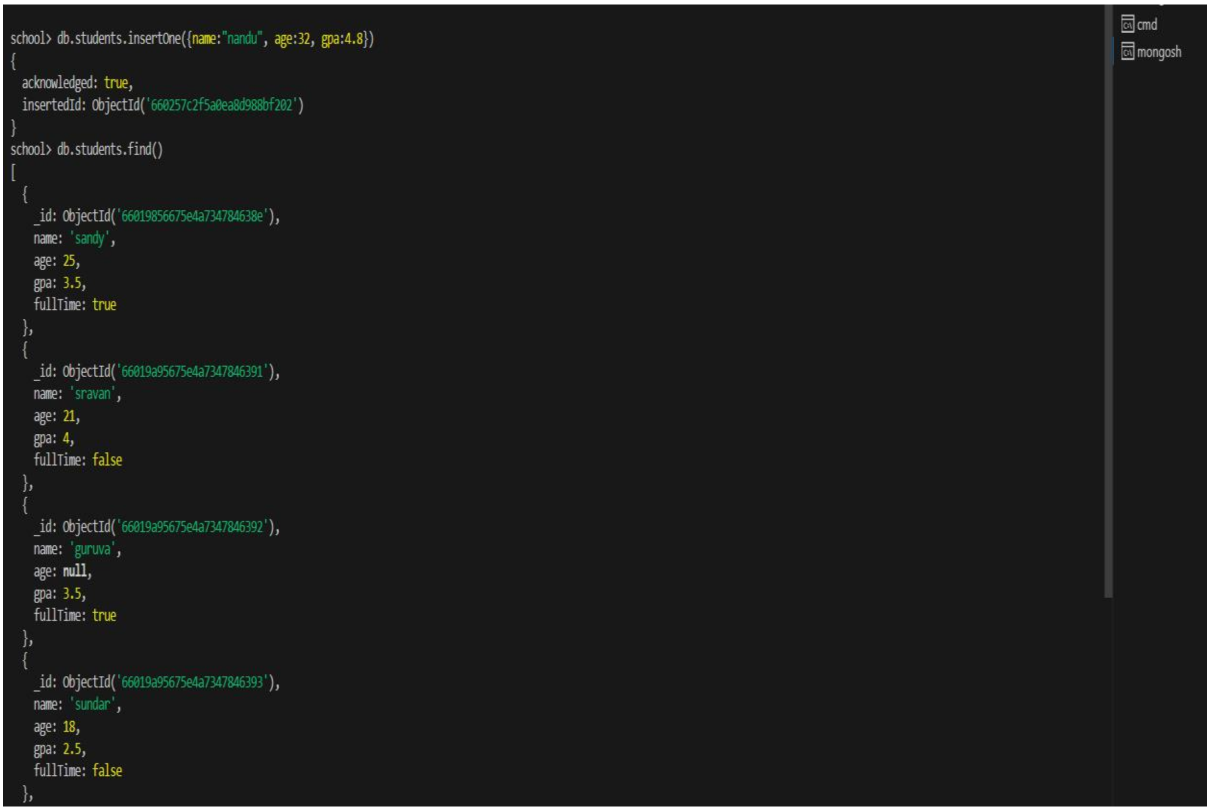
→Database Setup: Created a new MongoDB database called “School”.

→Collection Creation: Created a collection named as “students” within the school database.

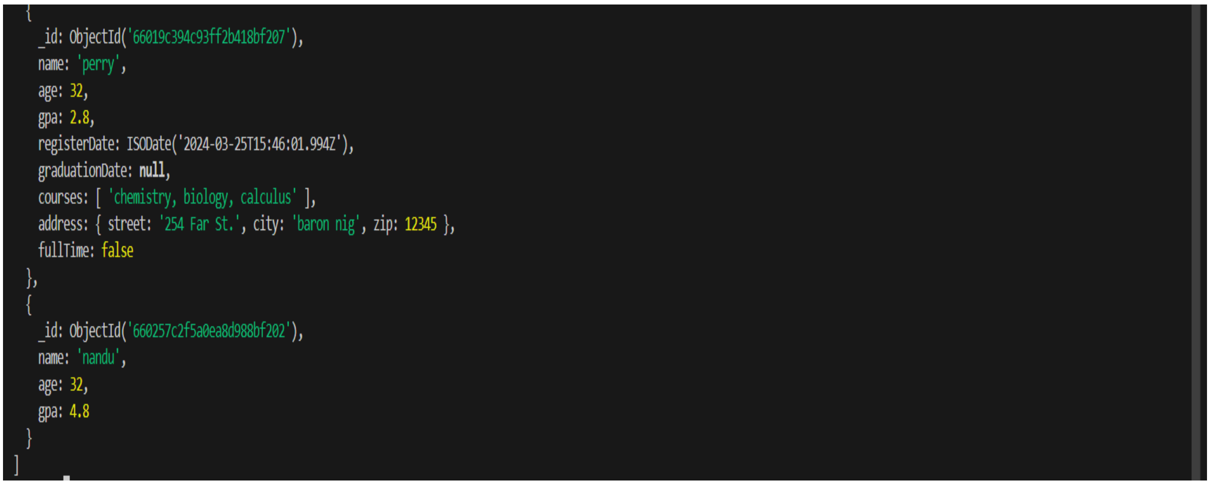


→Document Insertion: Inserted three documents into the Student collection, each representing a students with fields such as name, age and gpa.

*In MongoDB PowerShell:

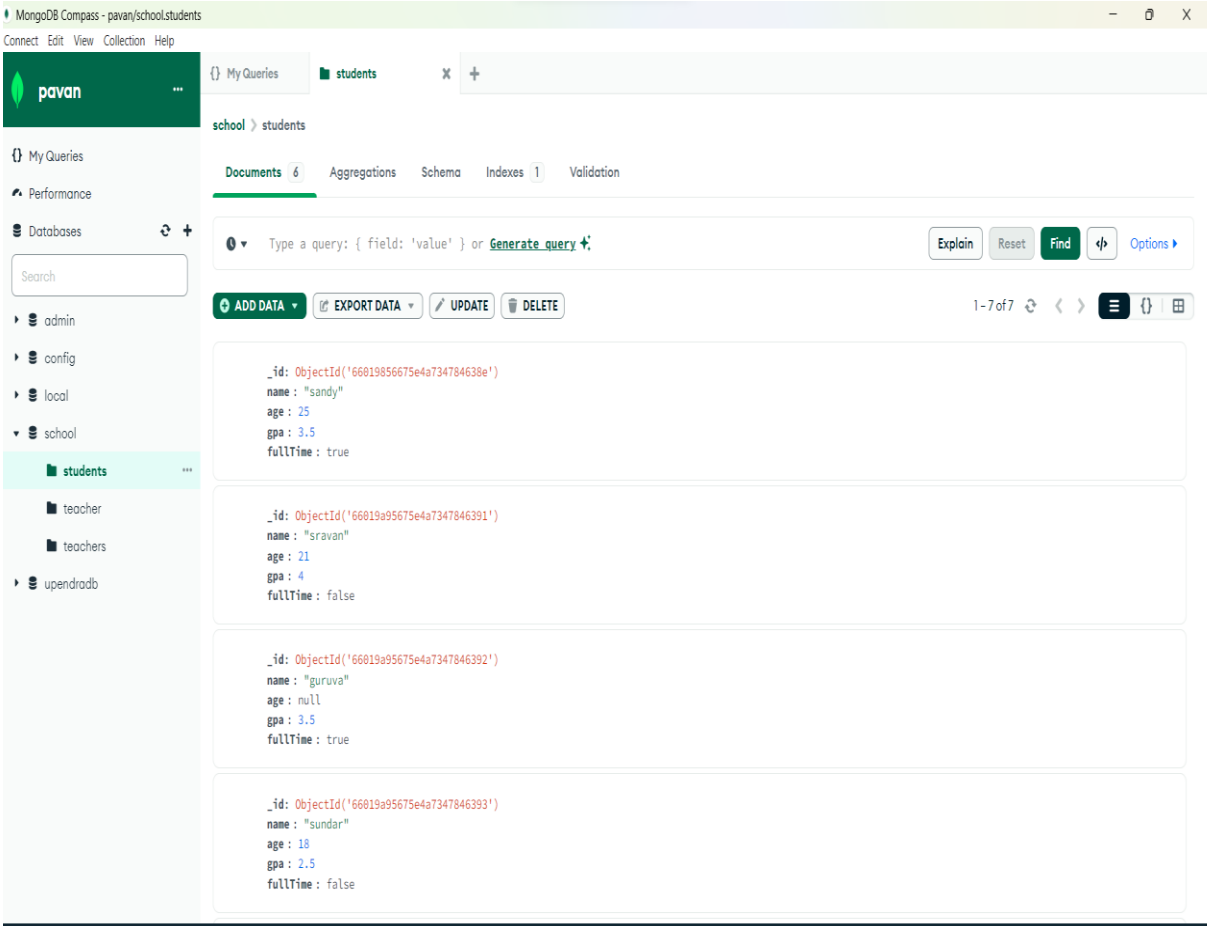


```
school> db.students.insertOne({name:"nandu", age:32, gpa:4.8})
{
  acknowledged: true,
  insertedId: ObjectId('660257c2f5a0ea8d988bf202')
}
school> db.students.find()
[
  {
    _id: ObjectId('66019856675e4a734784638e'),
    name: 'sandy',
    age: 25,
    gpa: 3.5,
    fullTime: true
  },
  {
    _id: ObjectId('66019a95675e4a7347846391'),
    name: 'sravan',
    age: 21,
    gpa: 4,
    fullTime: false
  },
  {
    _id: ObjectId('66019a95675e4a7347846392'),
    name: 'guruva',
    age: null,
    gpa: 3.5,
    fullTime: true
  },
  {
    _id: ObjectId('66019a95675e4a7347846393'),
    name: 'sundar',
    age: 18,
    gpa: 2.5,
    fullTime: false
  },
```



```
  {
    _id: ObjectId('66019c394c93ff2b418bf207'),
    name: 'perry',
    age: 32,
    gpa: 2.8,
    registerDate: ISODate('2024-03-25T15:46:01.994Z'),
    graduationDate: null,
    courses: [ 'chemistry, biology, calculus' ],
    address: { street: '254 Far St.', city: 'baron nig', zip: 12345 },
    fullTime: false
  },
  {
    _id: ObjectId('660257c2f5a0ea8d988bf202'),
    name: 'nandu',
    age: 32,
    gpa: 4.8
  }
]
```

In compass:

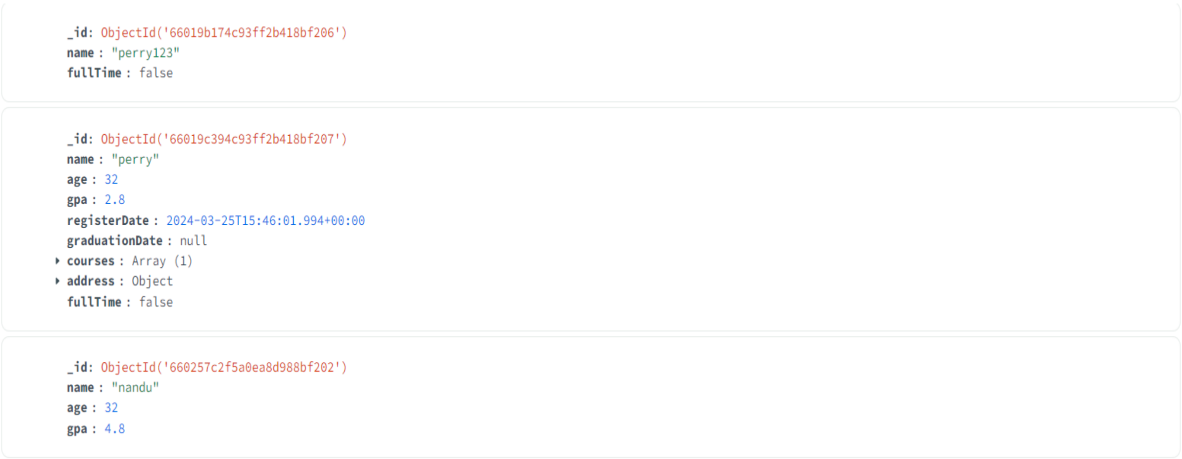


```
_id: ObjectId('66019856675e4a734784638e')
name : "sandy"
age : 25
gpa : 3.5
fullTime : true
```

```
_id: ObjectId('66019a95675e4a7347846391')
name : "sravan"
age : 21
gpa : 4
fullTime : false
```

```
_id: ObjectId('66019a95675e4a7347846392')
name : "guruva"
age : null
gpa : 3.5
fullTime : true
```

```
_id: ObjectId('66019a95675e4a7347846393')
name : "sundar"
age : 18
gpa : 2.5
fullTime : false
```



```
_id: ObjectId('66019b174c93ff2b418bf206')
name : "perry123"
fullTime : false
```

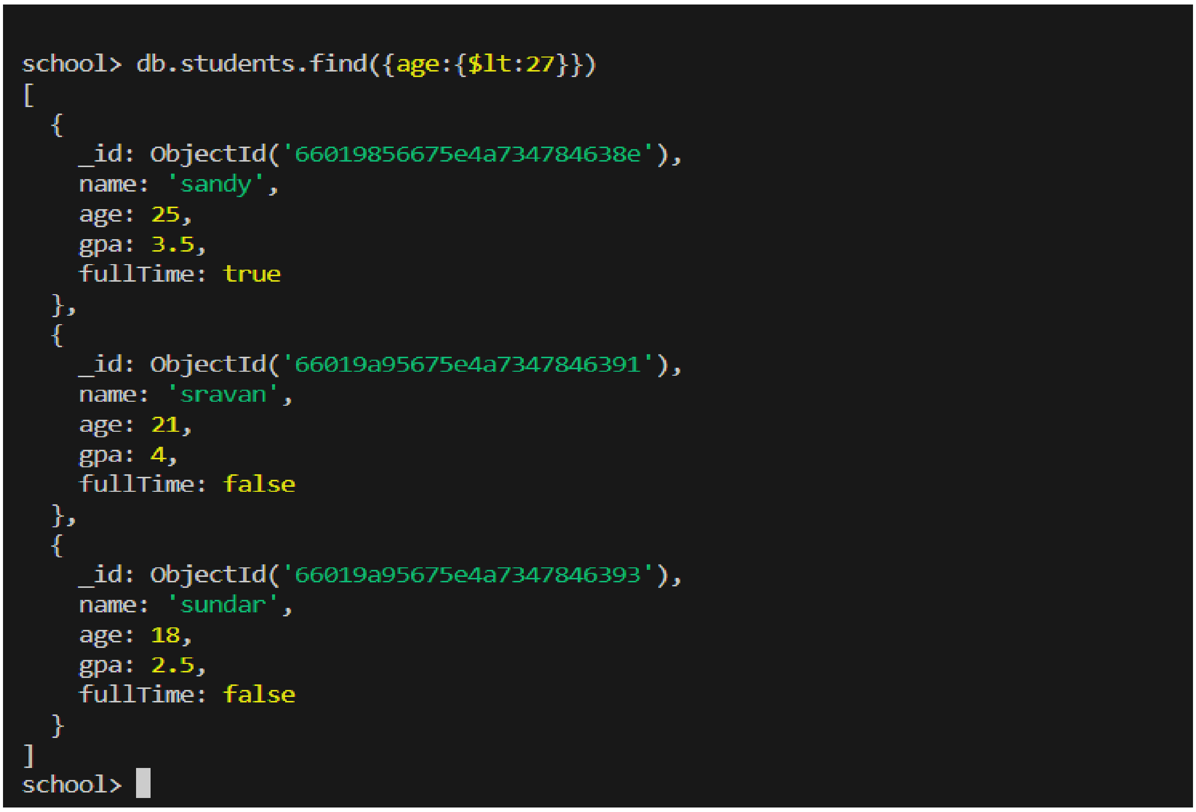
```
_id: ObjectId('66019c394c93ff2b418bf207')
name : "perry"
age : 32
gpa : 2.8
registerDate : 2024-03-25T15:46:01.994+00:00
graduationDate : null
courses : Array (1)
address : Object
fullTime : false
```

```
_id: ObjectId('660257c2f5a0ea8d988bf202')
name : "nandu"
age : 32
gpa : 4.8
```

→<u>Quering</u>: Queries to retrieve: All students from the student collection. Students with age greater than or equal to 30.

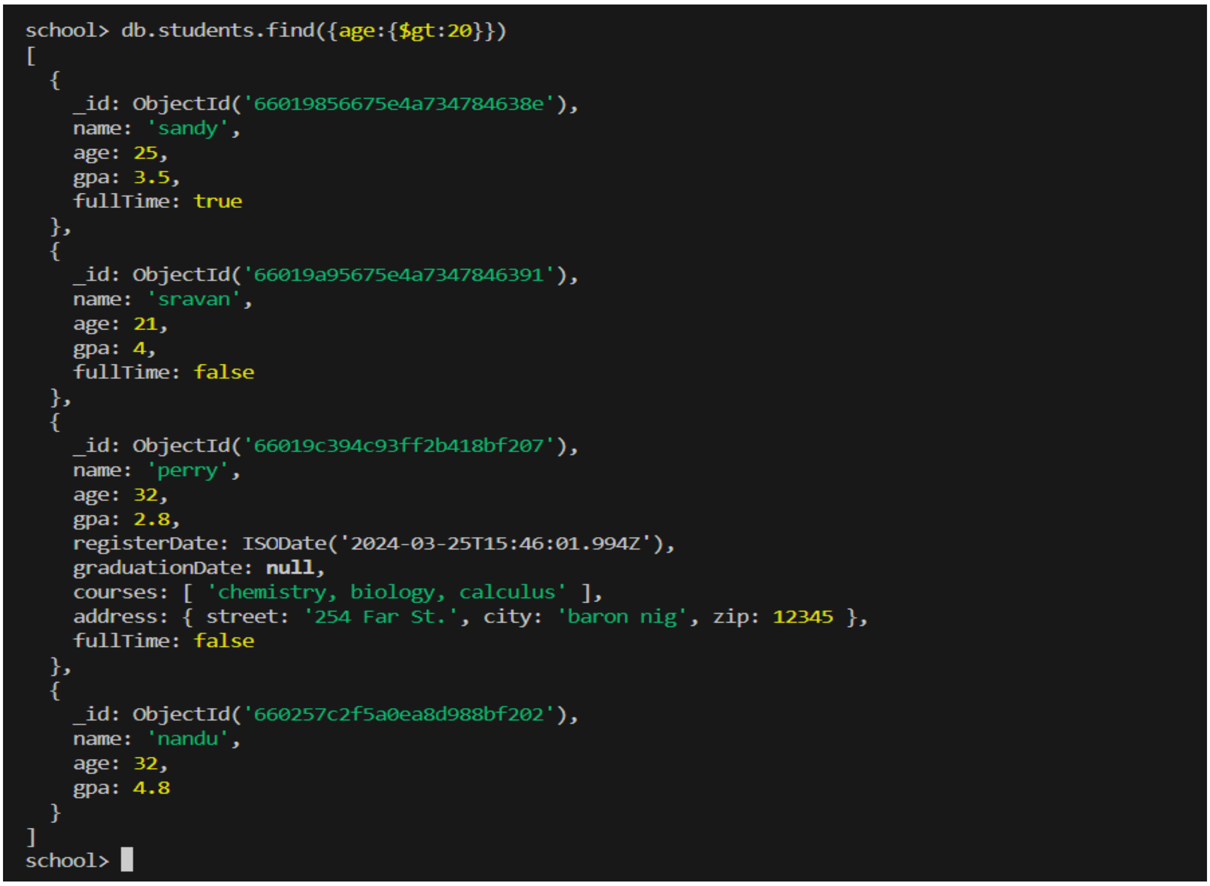
<u>*Comparison operator*</u>: Operators return data based on value comparisons

Less than: ($lt)



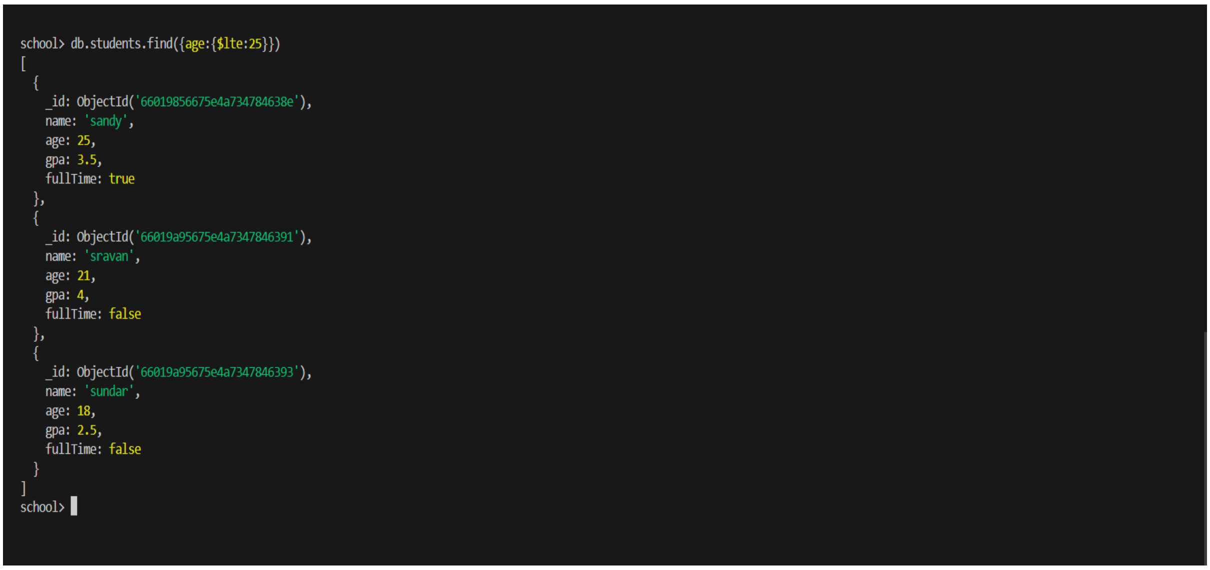
```
school> db.students.find({age:{$lt:27}})
[
  {
    _id: ObjectId('66019856675e4a734784638e'),
    name: 'sandy',
    age: 25,
    gpa: 3.5,
    fullTime: true
  },
  {
    _id: ObjectId('66019a95675e4a7347846391'),
    name: 'sravan',
    age: 21,
    gpa: 4,
    fullTime: false
  },
  {
    _id: ObjectId('66019a95675e4a7347846393'),
    name: 'sundar',
    age: 18,
    gpa: 2.5,
    fullTime: false
  }
]
school> 
```

Greater than: ($gt)



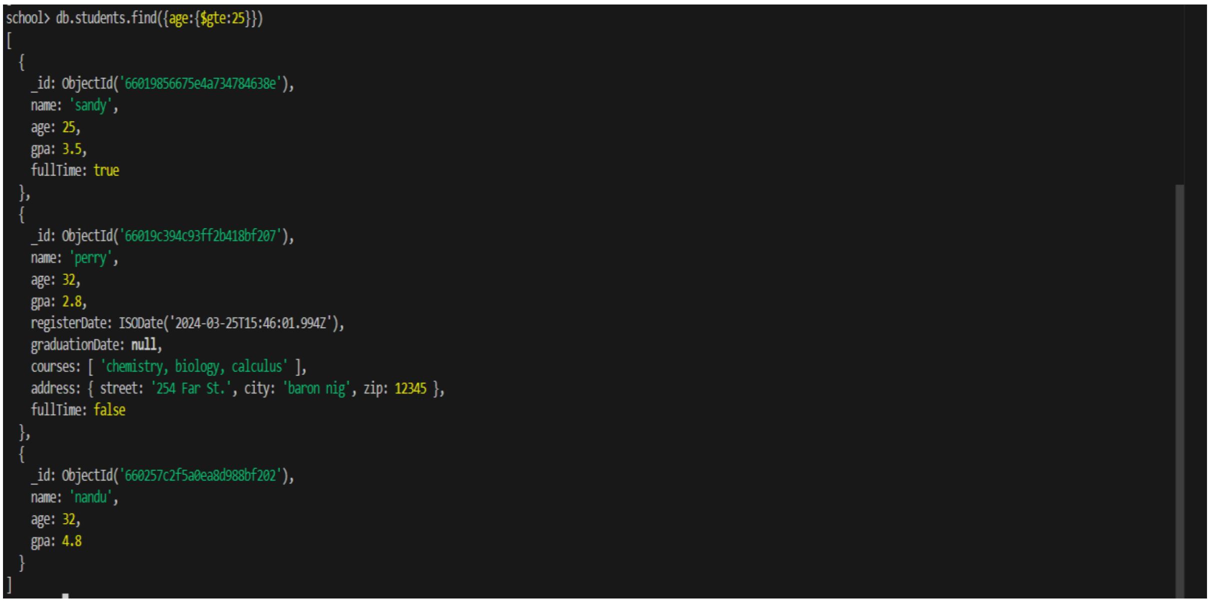
```
school> db.students.find({age:{$gt:20}})
[
  {
    _id: ObjectId('66019856675e4a734784638e'),
    name: 'sandy',
    age: 25,
    gpa: 3.5,
    fullTime: true
  },
  {
    _id: ObjectId('66019a95675e4a7347846391'),
    name: 'sravan',
    age: 21,
    gpa: 4,
    fullTime: false
  },
  {
    _id: ObjectId('66019c394c93ff2b418bf207'),
    name: 'perry',
    age: 32,
    gpa: 2.8,
    registerDate: ISODate('2024-03-25T15:46:01.994Z'),
    graduationDate: null,
    courses: [ 'chemistry, biology, calculus' ],
    address: { street: '254 Far St.', city: 'baron nig', zip: 12345 },
    fullTime: false
  },
  {
    _id: ObjectId('660257c2f5a0ea8d988bf202'),
    name: 'nandu',
    age: 32,
    gpa: 4.8
  }
]
school> 
```

Less than equal: ($lte)



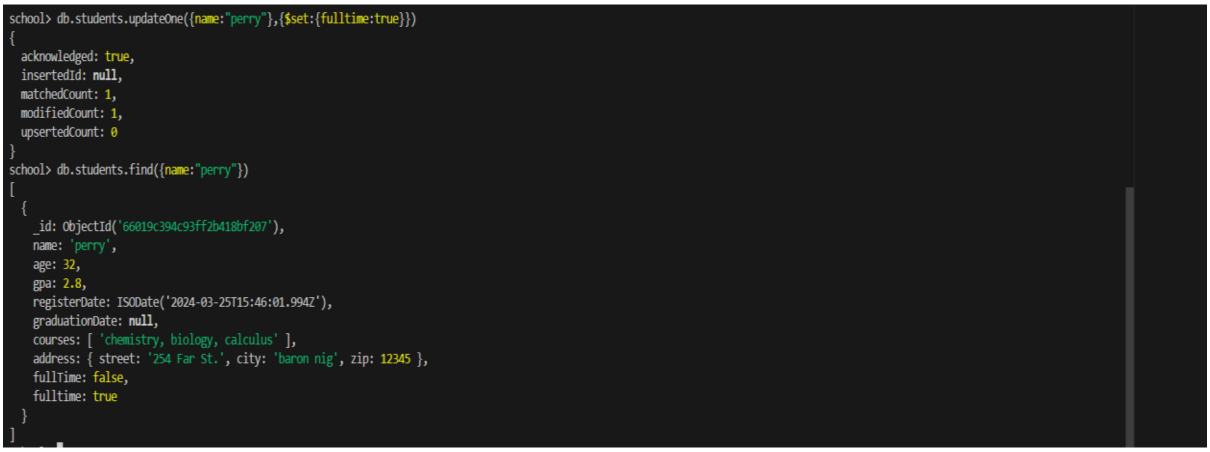
```
school> db.students.find({age:{$lte:25}})
[
  {
    _id: ObjectId('66019856675e4a734784638e'),
    name: 'sandy',
    age: 25,
    gpa: 3.5,
    fullTime: true
  },
  {
    _id: ObjectId('66019a95675e4a7347846391'),
    name: 'sravan',
    age: 21,
    gpa: 4,
    fullTime: false
  },
  {
    _id: ObjectId('66019a95675e4a7347846393'),
    name: 'sundar',
    age: 18,
    gpa: 2.5,
    fullTime: false
  }
]
school> 
```

Greater than equal: ($gte)

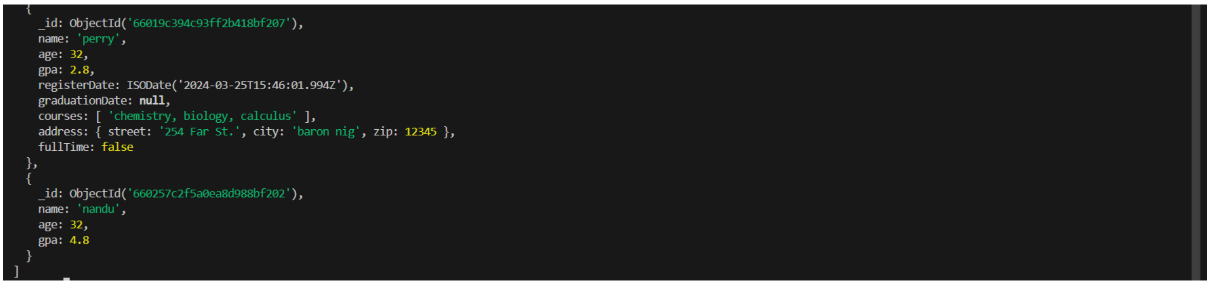


```
school> db.students.find({age:{$gte:25}})
[
  {
    _id: ObjectId('66019856675e4a734784638e'),
    name: 'sandy',
    age: 25,
    gpa: 3.5,
    fullTime: true
  },
  {
    _id: ObjectId('66019c394c93ff2b418bf207'),
    name: 'perry',
    age: 32,
    gpa: 2.8,
    registerDate: ISODate('2024-03-25T15:46:01.994Z'),
    graduationDate: null,
    courses: [ 'chemistry, biology, calculus' ],
    address: { street: '254 Far St.', city: 'baron nig', zip: 12345 },
    fullTime: false
  },
  {
    _id: ObjectId('660257c2f5a0ea8d988bf202'),
    name: 'nandu',
    age: 32,
    gpa: 4.8
  }
]
```

*Update operator: Updated the job of a student with a specific name.

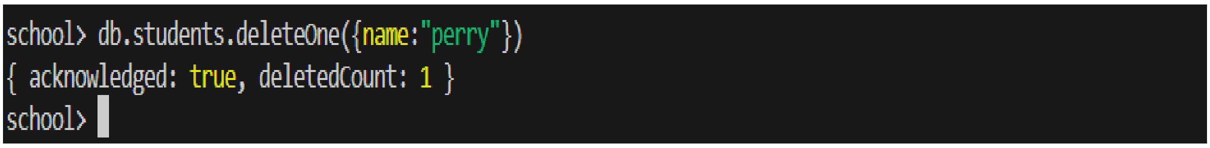


```
school> db.students.updateOne({name:"perry"},{$set:{fulltime:true}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
school> db.students.find({name:"perry"})
[
  {
    _id: ObjectId('66019c394c93ff2b418bf207'),
    name: 'perry',
    age: 32,
    gpa: 2.8,
    registerDate: ISODate('2024-03-25T15:46:01.994Z'),
    graduationDate: null,
    courses: [ 'chemistry, biology, calculus' ],
    address: { street: '254 Far St.', city: 'baron nig', zip: 12345 },
    fullTime: false,
    fulltime: true
  }
]
```



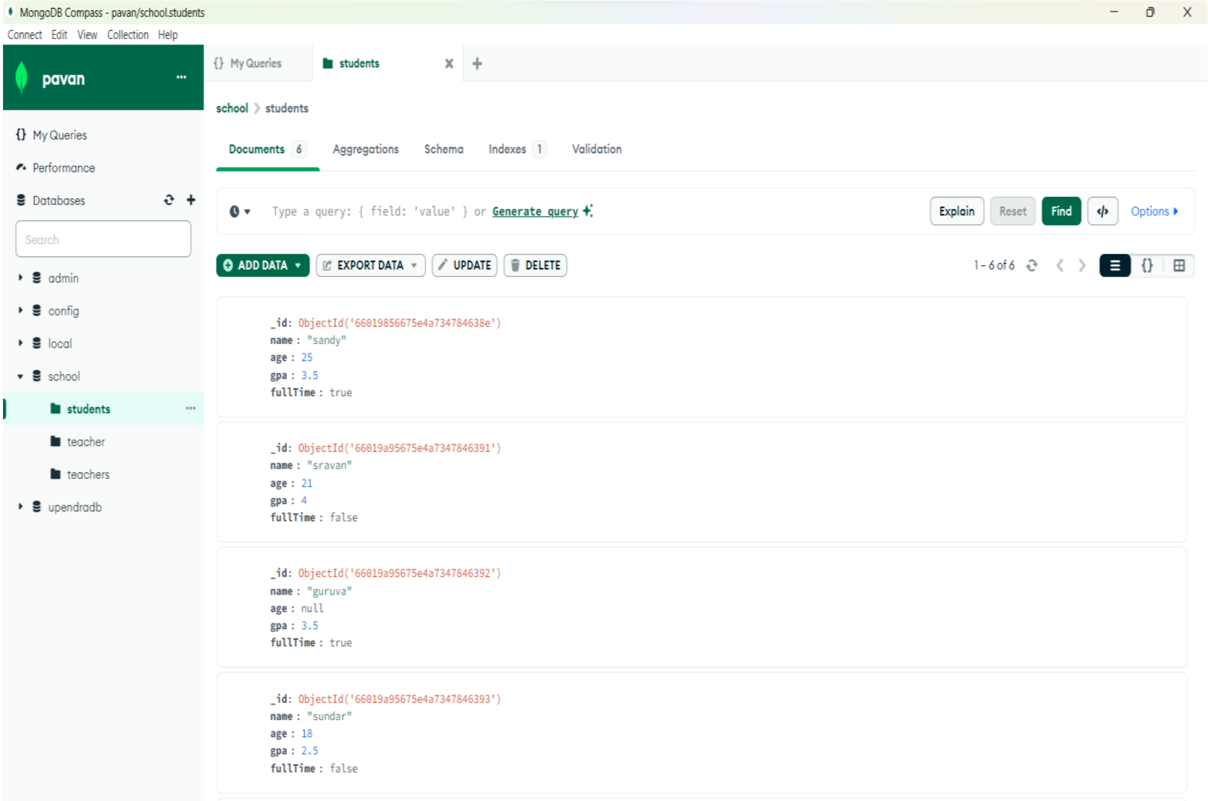
```
  {
    _id: ObjectId('66019c394c93ff2b418bf207'),
    name: 'perry',
    age: 32,
    gpa: 2.8,
    registerDate: ISODate('2024-03-25T15:46:01.994Z'),
    graduationDate: null,
    courses: [ 'chemistry, biology, calculus' ],
    address: { street: '254 Far St.', city: 'baron nig', zip: 12345 },
    fullTime: false
  },
  {
    _id: ObjectId('660257c2f5a0ea8d988bf202'),
    name: 'nandu',
    age: 32,
    gpa: 4.8
  }
]
```

*Deletion Operation: Delete a user document based on a specific email address.



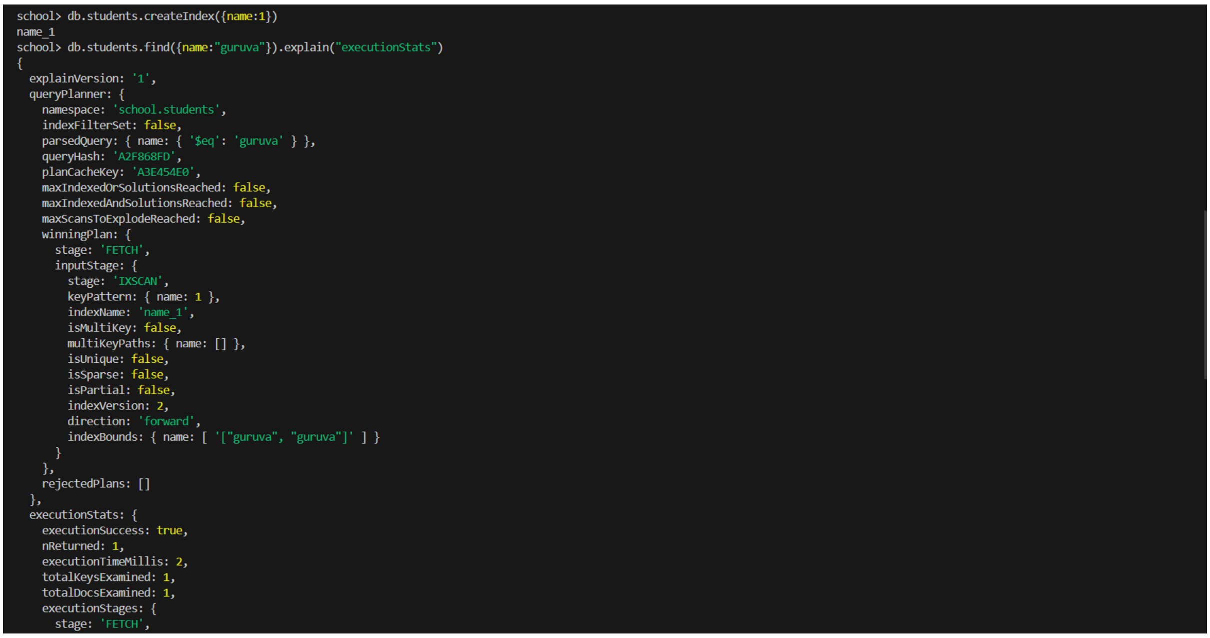
```
school> db.students.deleteOne({name:"perry"})
{ acknowledged: true, deletedCount: 1 }
school>
```

In compass:

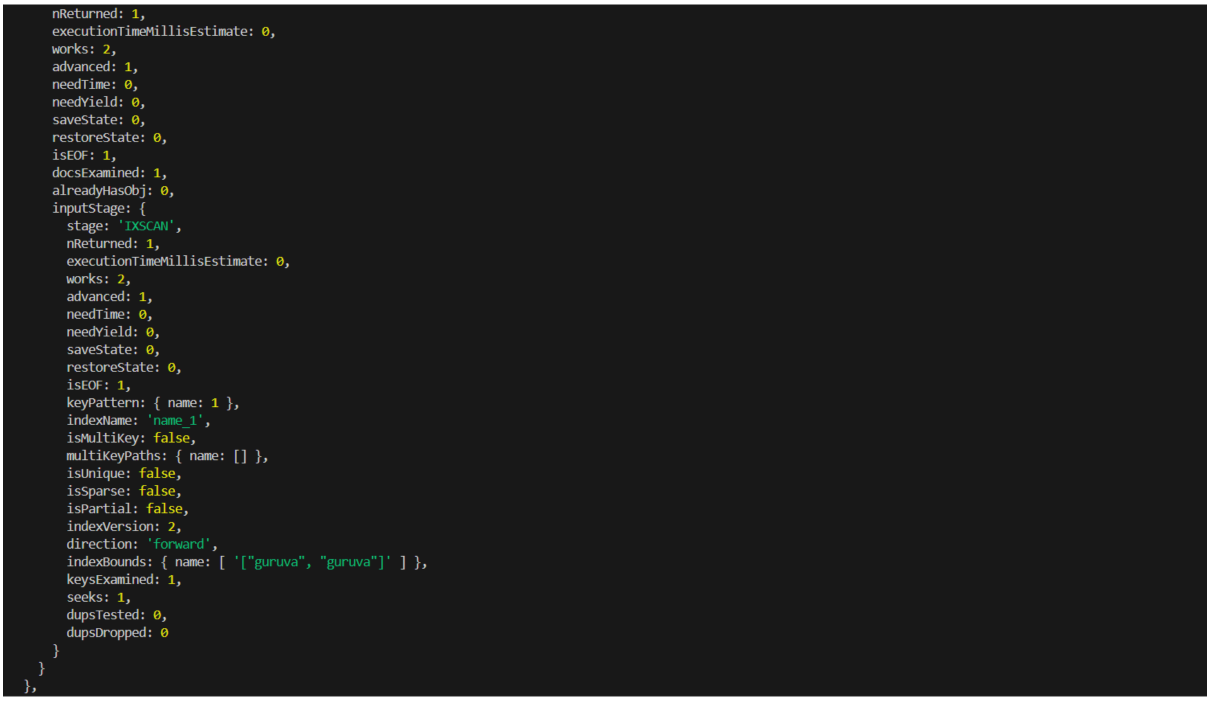


```
_id: ObjectId('66019856675e4a734784638e')
name : "sandy"
age : 25
gpa : 3.5
fullTime : true

_id: ObjectId('66019a95675e4a7347846391')
name : "sravan"
age : 21
gpa : 4
fullTime : false

_id: ObjectId('66019a95675e4a7347846392')
name : "guruva"
age : null
gpa : 3.5
fullTime : true

_id: ObjectId('66019a95675e4a7347846393')
name : "sundar"
age : 18
gpa : 2.5
fullTime : false
```

*Index Creation: Indexes support efficient execution of queries in MongoDB. Without indexes, MongoDB must scan every document in a collection to return query results. If an appropriate index exists for a query, MongoDB uses the index to limit the number of documents it must scan.



```
school> db.students.createIndex({name:1})
name_1
school> db.students.find({name:"guruva"}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'school.students',
    indexFilterSet: false,
    parsedQuery: { name: { '$eq': 'guruva' } },
    queryHash: 'A2F868FD',
    planCacheKey: 'A3E454E0',
    maxIndexedOrSolutionsReached: false,
    maxIndexedAndSolutionsReached: false,
    maxScansToExplodeReached: false,
    winningPlan: {
      stage: 'FETCH',
      inputStage: {
        stage: 'IXSCAN',
        keyPattern: { name: 1 },
        indexName: 'name_1',
        isMultiKey: false,
        multiKeyPaths: { name: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { name: [ '["guruva", "guruva"]' ] }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 1,
    executionTimeMillis: 2,
    totalKeysExamined: 1,
    totalDocsExamined: 1,
    executionStages: {
      stage: 'FETCH',
```



```
      nReturned: 1,
      executionTimeMillisEstimate: 0,
      works: 2,
      advanced: 1,
      needTime: 0,
      needYield: 0,
      saveState: 0,
      restoreState: 0,
      isEOF: 1,
      docsExamined: 1,
      alreadyHasObj: 0,
      inputStage: {
        stage: 'IXSCAN',
        nReturned: 1,
        executionTimeMillisEstimate: 0,
        works: 2,
        advanced: 1,
        needTime: 0,
        needYield: 0,
        saveState: 0,
        restoreState: 0,
        isEOF: 1,
        keyPattern: { name: 1 },
        indexName: 'name_1',
        isMultiKey: false,
        multiKeyPaths: { name: [] },
        isUnique: false,
        isSparse: false,
        isPartial: false,
        indexVersion: 2,
        direction: 'forward',
        indexBounds: { name: [ '["guruva", "guruva"]' ] },
        keysExamined: 1,
        seeks: 1,
        dupsTested: 0,
        dupsDropped: 0
      }
    }
  },
```