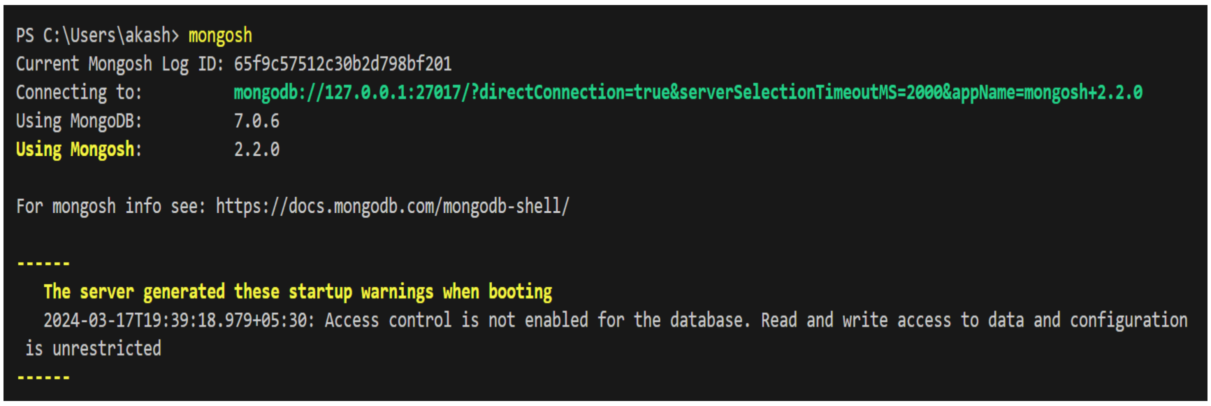
-Assignment: Creating a Database Using MongoDB and Mongosh

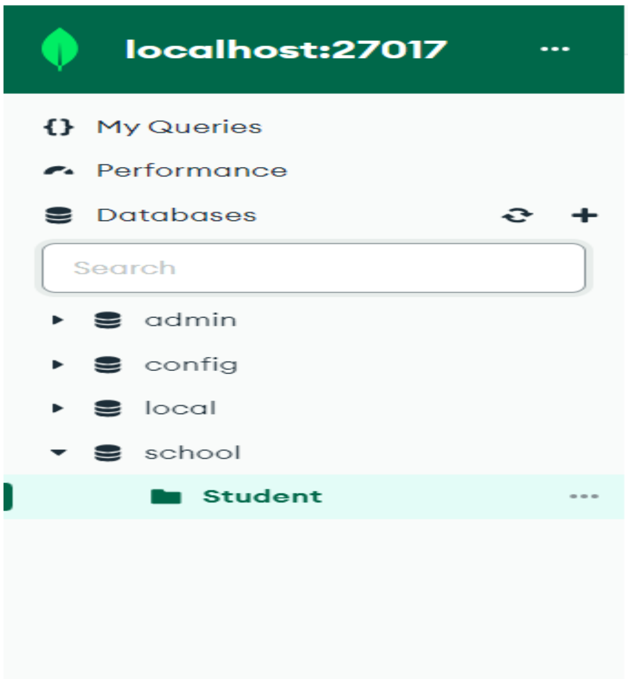
MongoDB: MongoDB is a source-available, cross-platform, document-oriented database program. Classified as a NoSQL database product, MongoDB utilizes JSON-like documents with optional schemas. MongoDB is developed by MongoDB Inc.



```
PS C:\Users\akash> mongosh
Current Mongosh Log ID: 65f9c57512c30b2d798bf201
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.0
Using MongoDB:          7.0.6
Using Mongosh:          2.2.0

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

------
   The server generated these startup warnings when booting
   2024-03-17T19:39:18.979+05:30: Access control is not enabled for the database. Read and write access to data and configuration
 is unrestricted
------
```

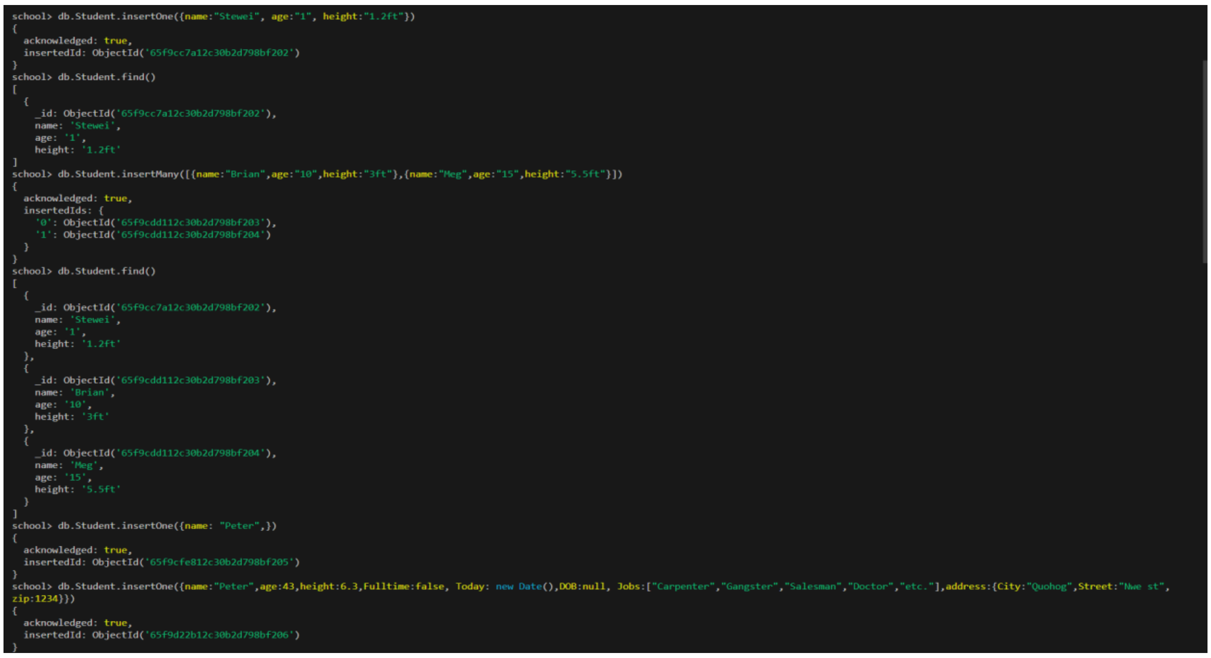
Assignment Details:

- **Database Setup**: Created a new MongoDB database called "School".
- **Collection Creation**: Created a collection named Student within the school database.



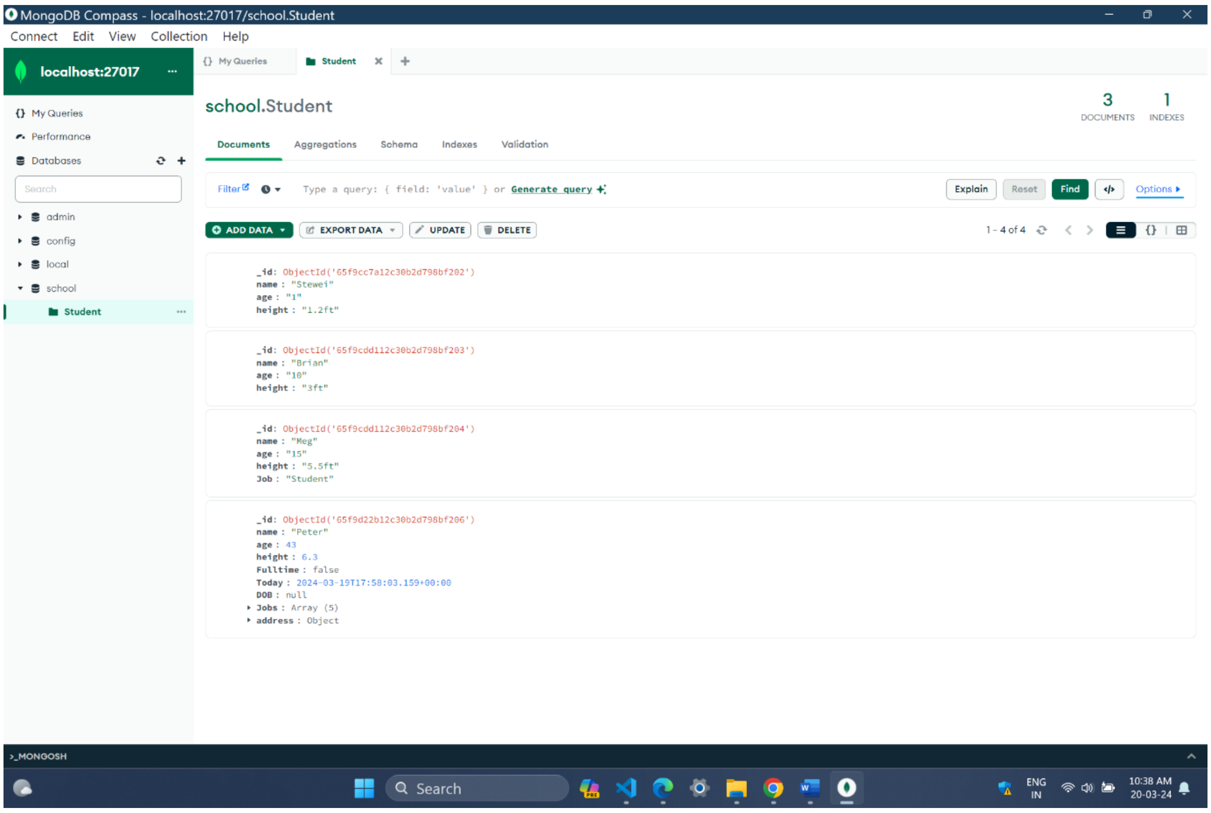
- **<u>Document Insertion</u>**: Inserted three documents into the Student collection, each representing a Student with fields such as name, age, and height.

<u>In MongoDB PowerShell:</u>



```
school> db.Student.insertOne({name:"Stewei", age:"1", height:"1.2ft"})
{
  acknowledged: true,
  insertedId: ObjectId('65f9cc7a12c30b2d798bf202')
}
school> db.Student.find()
[
  {
    _id: ObjectId('65f9cc7a12c30b2d798bf202'),
    name: 'Stewei',
    age: '1',
    height: '1.2ft'
]
school> db.Student.insertMany([{name:"Brian",age:"10",height:"3ft"},{name:"Meg",age:"15",height:"5.5ft"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65f9cdd112c30b2d798bf203'),
    '1': ObjectId('65f9cdd112c30b2d798bf204')
  }
}
school> db.Student.find()
[
  {
    _id: ObjectId('65f9cc7a12c30b2d798bf202'),
    name: 'Stewei',
    age: '1',
    height: '1.2ft'
  },
  {
    _id: ObjectId('65f9cdd112c30b2d798bf203'),
    name: 'Brian',
    age: '10',
    height: '3ft'
  },
  {
    _id: ObjectId('65f9cdd112c30b2d798bf204'),
    name: 'Meg',
    age: '15',
    height: '5.5ft'
  }
]
school> db.Student.insertOne({name: "Peter",})
{
  acknowledged: true,
  insertedId: ObjectId('65f9cfe812c30b2d798bf205')
}
school> db.Student.insertOne({name:"Peter",age:43,height:6.3,Fulltime:false, Today: new Date(),DOB:null, Jobs:["Carpenter","Gangster","Salesman","Doctor","etc."],address:{City:"Quohog",Street:"Nwe st",
zip:1234}})
{
  acknowledged: true,
  insertedId: ObjectId('65f9d22b12c30b2d798bf206')
}
```

<u>In Compass:</u>

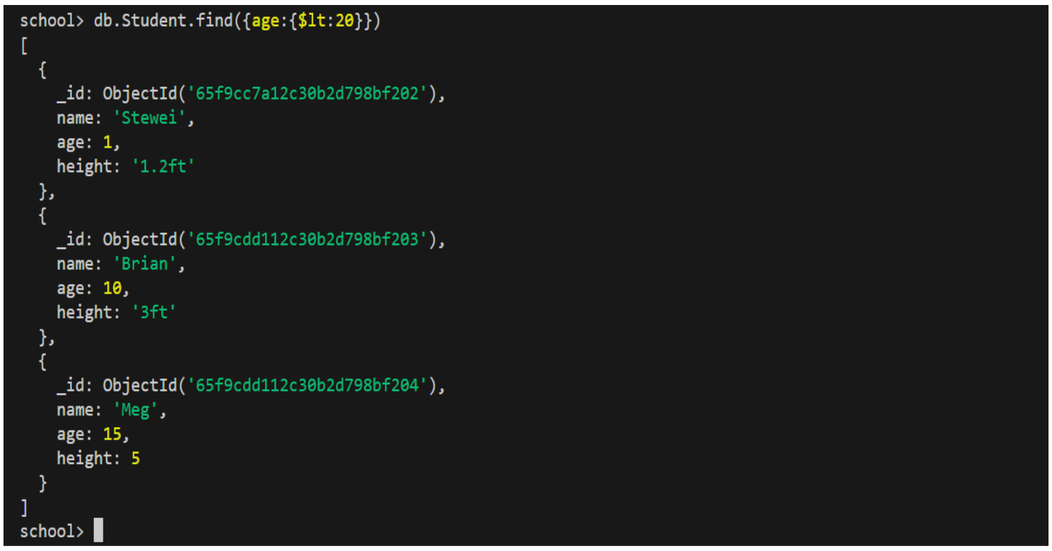

MongoDB Compass - localhost:27017/school.Student
Connect Edit View Collection Help
localhost:27017
My Queries
Student
My Queries
Performance
Databases
Search
admin
config
local
school
Student
school.Student
3 DOCUMENTS
1 INDEXES
Documents Aggregations Schema Indexes Validation
Filter
Type a query: { field: 'value' } or Generate query
Explain Reset Find Options
ADD DATA EXPORT DATA UPDATE DELETE
1 - 4 of 4
>_MONGOSH
Search
ENG IN
10:38 AM 20-03-24

```
_id: ObjectId('65f9cc7a12c30b2d798bf202')
name : "Stewei"
age : "1"
height : "1.2ft"

_id: ObjectId('65f9cdd112c30b2d798bf203')
name : "Brian"
age : "10"
height : "3ft"

_id: ObjectId('65f9cdd112c30b2d798bf204')
name : "Meg"
age : "15"
height : "5.5ft"
Job : "Student"

_id: ObjectId('65f9d22b12c30b2d798bf206')
name : "Peter"
age : 43
height : 6.3
Fulltime : false
Today : 2024-03-19T17:58:03.159+00:00
DOB : null
▸ Jobs : Array (5)
▸ address : Object
```

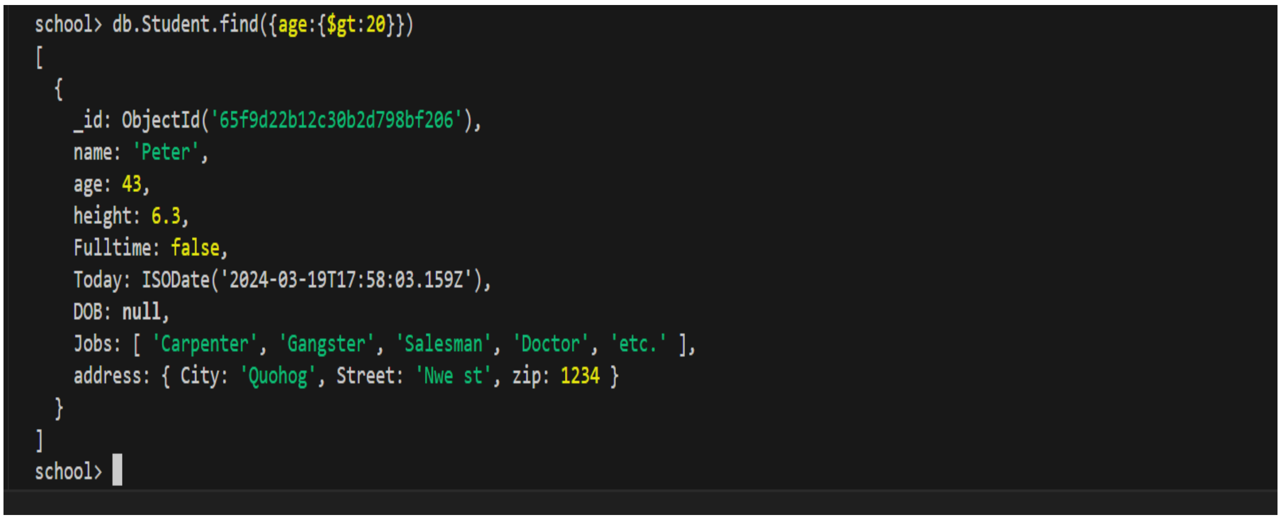
- **Querying:** queries to retrieve: All students from the Student collection. Student with an age greater than or equal to 30.

Comparison Operator: operators return data based on value comparisons

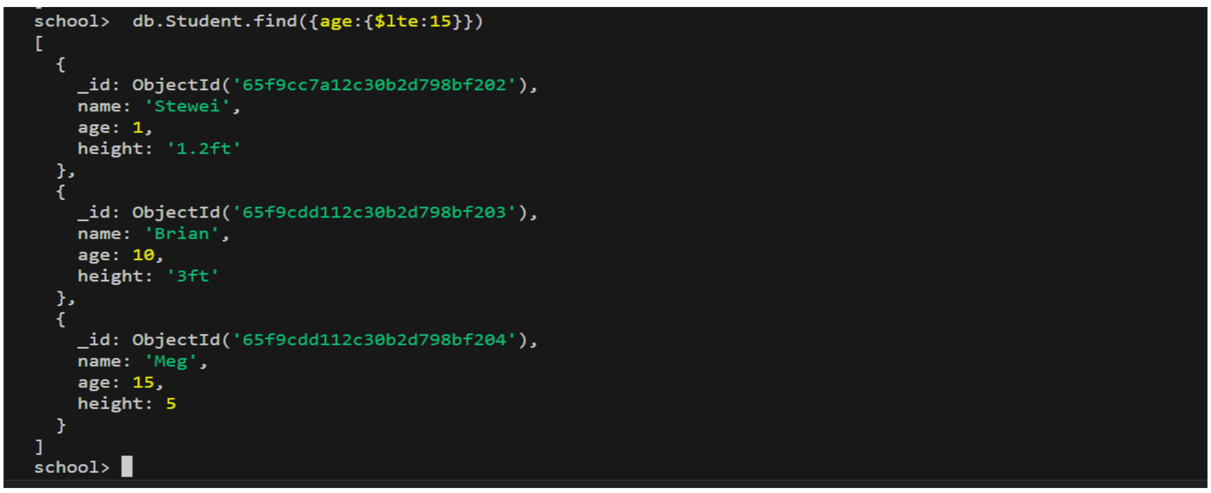
Less than: ($lt)


```
school> db.Student.find({age:{$lt:20}})
[
  {
    _id: ObjectId('65f9cc7a12c30b2d798bf202'),
    name: 'Stewei',
    age: 1,
    height: '1.2ft'
  },
  {
    _id: ObjectId('65f9cdd112c30b2d798bf203'),
    name: 'Brian',
    age: 10,
    height: '3ft'
  },
  {
    _id: ObjectId('65f9cdd112c30b2d798bf204'),
    name: 'Meg',
    age: 15,
    height: 5
  }
]
school> 
```

Greater than: ($gt)

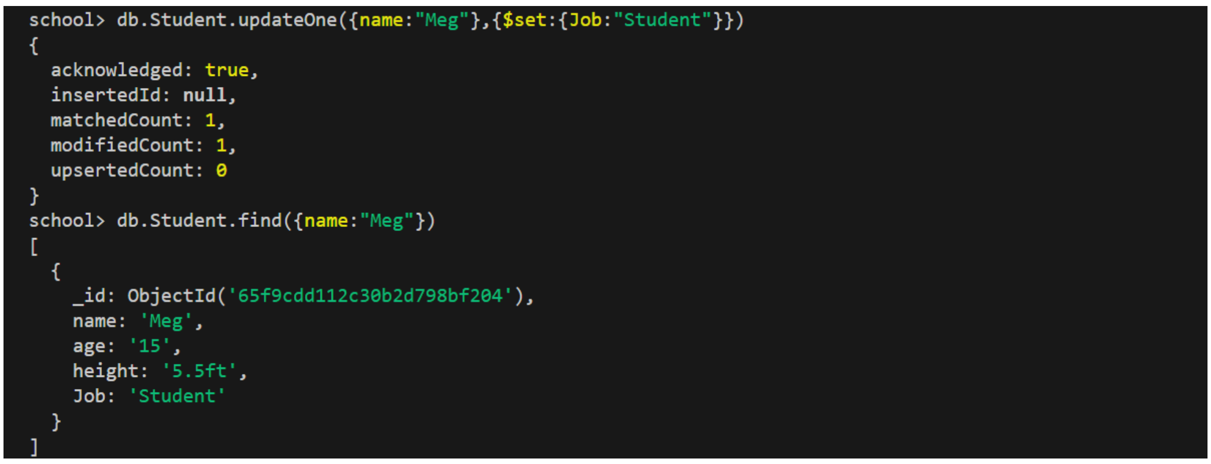

```
school> db.Student.find({age:{$gt:20}})
[
  {
    _id: ObjectId('65f9d22b12c30b2d798bf206'),
    name: 'Peter',
    age: 43,
    height: 6.3,
    Fulltime: false,
    Today: ISODate('2024-03-19T17:58:03.159Z'),
    DOB: null,
    Jobs: [ 'Carpenter', 'Gangster', 'Salesman', 'Doctor', 'etc.' ],
    address: { City: 'Quohog', Street: 'Nwe st', zip: 1234 }
  }
]
school> 
```

Less than equal ($lte)

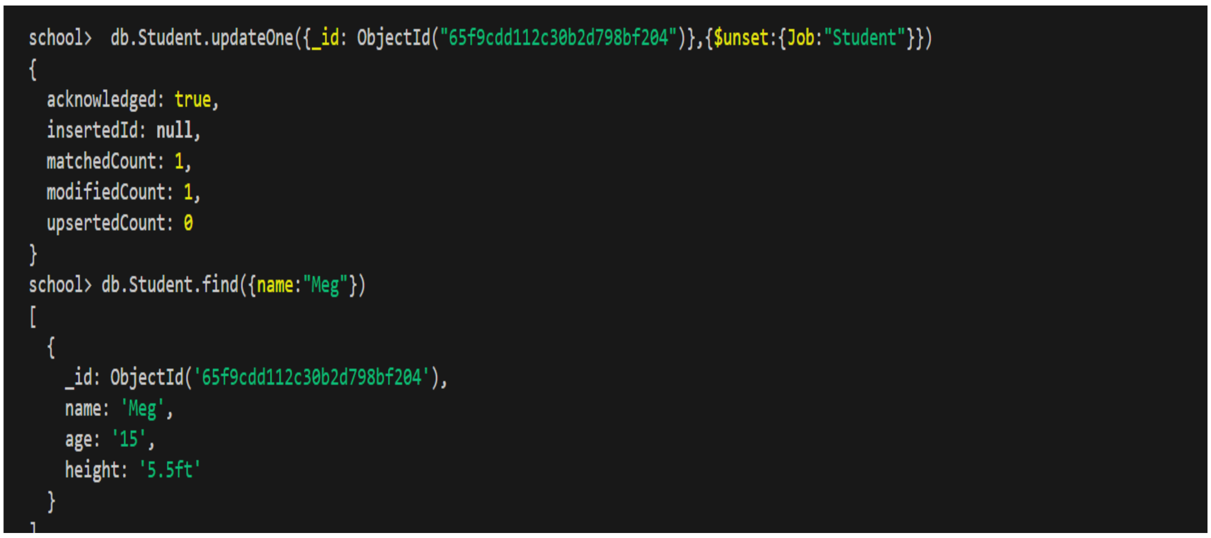


```
school>  db.Student.find({age:{$lte:15}})
[
  {
    _id: ObjectId('65f9cc7a12c30b2d798bf202'),
    name: 'Stewei',
    age: 1,
    height: '1.2ft'
  },
  {
    _id: ObjectId('65f9cdd112c30b2d798bf203'),
    name: 'Brian',
    age: 10,
    height: '3ft'
  },
  {
    _id: ObjectId('65f9cdd112c30b2d798bf204'),
    name: 'Meg',
    age: 15,
    height: 5
  }
]
school>
```

- <u>**Update Operation:**</u> Updated the Job of a student with a specific name.



```
school> db.Student.updateOne({name:"Meg"},{$set:{Job:"Student"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
school> db.Student.find({name:"Meg"})
[
  {
    _id: ObjectId('65f9cdd112c30b2d798bf204'),
    name: 'Meg',
    age: '15',
    height: '5.5ft',
    Job: 'Student'
  }
]
```



```
school>  db.Student.updateOne({_id: ObjectId("65f9cdd112c30b2d798bf204")},{$unset:{Job:"Student"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
school> db.Student.find({name:"Meg"})
[
  {
    _id: ObjectId('65f9cdd112c30b2d798bf204'),
    name: 'Meg',
    age: '15',
    height: '5.5ft'
  }
]
```

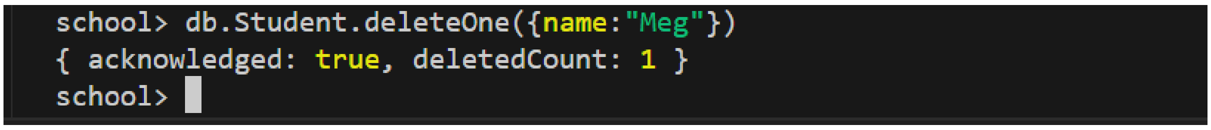
By default, the `db.collection.update()` method updates a **single** document. Include the option multi: true to update all documents that match the query criteria.

- **<u>Deletion Operation:</u>**

The MongoDB shell provides the following methods to delete documents from a collection:

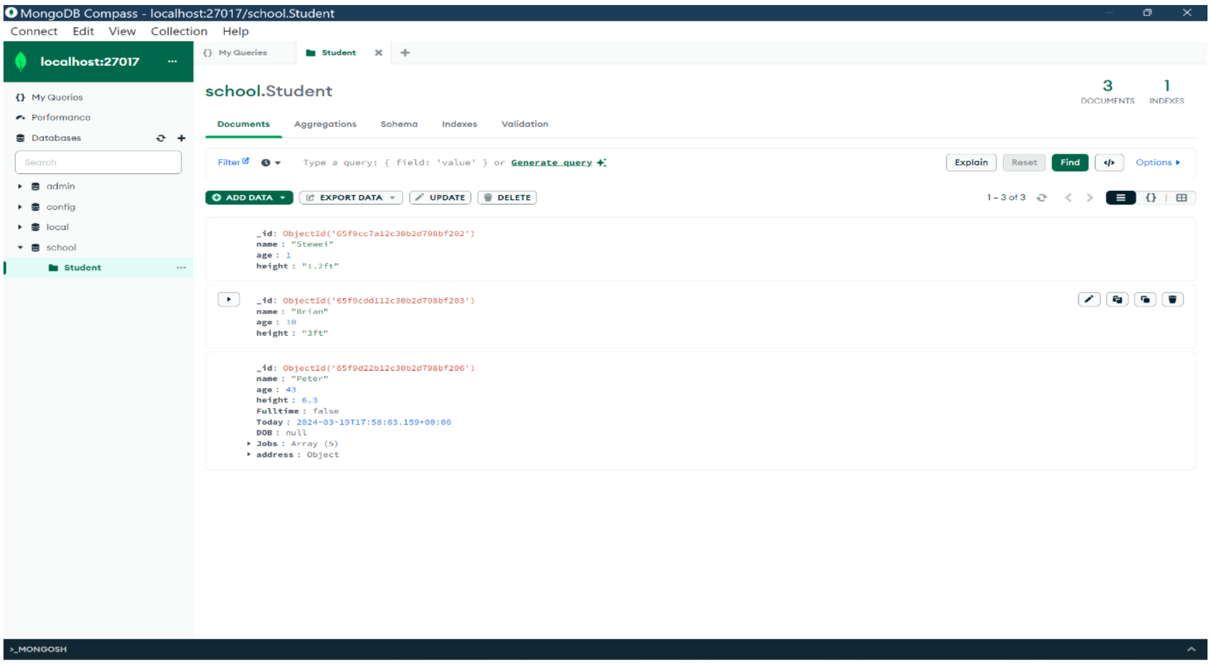
- To delete multiple documents, use `db.collection.deleteMany()`.
- To delete a single document, use `db.collection.deleteOne()`.

- Delete a user document based on a specific name.



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

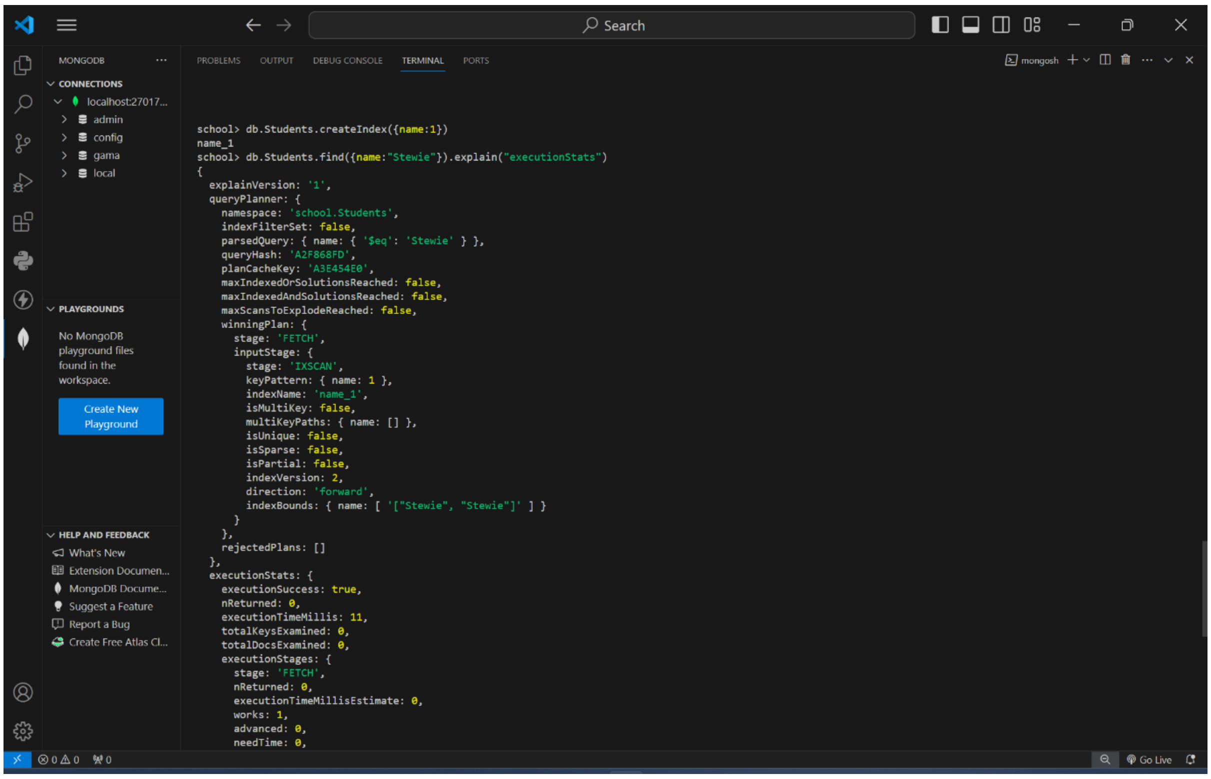
In compass:



```
_id: ObjectId('65f9cc7a12c30b2d798bf202')
name : "Stewei"
age : 1
height : "1.2ft"

_id: ObjectId('65f9cdd112c30b2d798bf203')
name : "Brian"
age : 10
height : "3ft"

_id: ObjectId('65f9d22b12c30b2d798bf206')
name : "Peter"
age : 43
height : 6.3
Fulltime : false
Today : 2024-03-19T17:58:03.159+00:00
DOB : null
▸ Jobs : Array (5)
▸ address : Object
```

- <u>**Index Creation:**</u>

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:"Stewie"}).explain("executionStats")
{
  explainVersion: '1',
  queryPlanner: {
    namespace: 'school.Students',
    indexFilterSet: false,
    parsedQuery: { name: { '$eq': 'Stewie' } },
    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: [ '["Stewie", "Stewie"]' ] }
      }
    },
    rejectedPlans: []
  },
  executionStats: {
    executionSuccess: true,
    nReturned: 0,
    executionTimeMillis: 11,
    totalKeysExamined: 0,
    totalDocsExamined: 0,
    executionStages: {
      stage: 'FETCH',
      nReturned: 0,
      executionTimeMillisEstimate: 0,
      works: 1,
      advanced: 0,
      needTime: 0,
```