Index are good for faster reading operations, but they are bad bro writing operations

Exercises:

Exercise 1: Inserting Documents

1. Create a database named "mydb".
2. Create a collection named "students".
3. Insert a few documents with fields like "name", "age", and "subject" for different students.

Exercise 2: Querying Documents

1. Find all documents in the "students" collection.
2. Find documents where the "age" is greater than 20.
3. Find documents where the "subject" is "Math".

Exercise 3: Updating Documents

1. Update the "age" of a student with a specific "name".
2. Add a new field "grade" with a value to all documents in the collection.

Exercise 4: Deleting Documents

1. Delete a document with a specific "name".
2. Delete all documents with "age" less than 18.

Exercise 5: Indexing

1. Create an index on the "name" field of the "students" collection.
2. Run a query that benefits from the index you created.

Exercise 6: Aggregation

1. Calculate the average age of all students.
2. Group students by subject and calculate the total count of students in each subject.

Exercise 7: Working with Nested Data

1. Create a collection named "restaurants".
2. Insert documents with fields like "name", "cuisine", "address", and nested arrays like "grades".
3. Find restaurants that have received a grade of "A".

MongoDB Database References, Model Tree Structures, MongoDB Analyzing Queries, MongoDB Atomic Operations, MongoDB Map Reduce, MongoDB Text Search, MongoDB Regular Expression, MongoDB Capped Collections, Data Storage, Working with Datatypes, Collections, Document Datatypes, Creating \_id Fields

1. Create a new Database called library  
2. Implement a whole library using various insert, update methods with relevant fields in each  
3. Write queries for the following :   
1. list members with no book borrows  
2. List top 10 members by highest borrowing

Solution:

school> show dbs

Library 184.00 KiB

admin 40.00 KiB

config 112.00 KiB

local 72.00 KiB

school> use school

already on db school

school> use Library

switched to db Library

Library> use school

switched to db school

school> show dbs

Library 184.00 KiB

admin 40.00 KiB

config 120.00 KiB

local 72.00 KiB

school 8.00 KiB

school> use school

already on db school

school> show dbs

Library 184.00 KiB

admin 40.00 KiB

config 120.00 KiB

local 72.00 KiB

school 8.00 KiB

school> show collections

students

school> use students

switched to db students

students> db.students.insertOne({name:"Chris"})

{

acknowledged: true,

insertedId: ObjectId("64de02cba17b4ed899070120")

}

students> db.students.find()

[ { \_id: ObjectId("64de02cba17b4ed899070120"), name: 'Chris' } ]

students> db.students.updateOne({name:"Chris"},{$set: {age:20})

Uncaught:

SyntaxError: Unexpected token, expected "," (1:52)

> 1 | db.students.updateOne({name:"Chris"},{$set: {age:20})

| ^

2 |

students> db.students.updateOne({name:"Chris"},{$set: {age:20}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

students> db.students.find()

[

{ \_id: ObjectId("64de02cba17b4ed899070120"), name: 'Chris', age: 20 }

]

students> db.students.updateOne({name:"Chris"},{$set: {subject:"Maths"}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

students> db.students.find()

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 20,

subject: 'Maths'

}

]

students> db.students.updateMany({name:"Mody", age:18, subject:"Physics"})

MongoInvalidArgumentError: Update document requires atomic operators

students> db.students.updateMany([{name:"Mody", age:18, subject:"Physics"}])

MongoInvalidArgumentError: Update document requires atomic operators

students> db.students.insertMany([{name:"Mody", age:18, subject:"Physics"}])

{

acknowledged: true,

insertedIds: { '0': ObjectId("64de03f3a17b4ed899070121") }

}

students> db.students.find()

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 20,

subject: 'Maths'

},

{

\_id: ObjectId("64de03f3a17b4ed899070121"),

name: 'Mody',

age: 18,

subject: 'Physics'

}

]

students> db.students.insertMany([{name:"Gary", age:22, subject:"chemistry"}])

{

acknowledged: true,

insertedIds: { '0': ObjectId("64de040da17b4ed899070122") }

}

students> db.students.insertMany([{name:"Layla", age:25, subject:"Geology"}])

{

acknowledged: true,

insertedIds: { '0': ObjectId("64de0424a17b4ed899070123") }

}

students> db.students.find()

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 20,

subject: 'Maths'

},

{

\_id: ObjectId("64de03f3a17b4ed899070121"),

name: 'Mody',

age: 18,

subject: 'Physics'

},

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology'

}

]

students> db.students.insertMany([{name:"Tom", age:28, subject:"Youtube"}])

{

acknowledged: true,

insertedIds: { '0': ObjectId("64de0453a17b4ed899070124") }

}

students> db.students.find()

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 20,

subject: 'Maths'

},

{

\_id: ObjectId("64de03f3a17b4ed899070121"),

name: 'Mody',

age: 18,

subject: 'Physics'

},

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology'

},

{

\_id: ObjectId("64de0453a17b4ed899070124"),

name: 'Tom',

age: 28,

subject: 'Youtube'

}

]

students> db.students.find({age:{$gt:20}})

[

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology'

},

{

\_id: ObjectId("64de0453a17b4ed899070124"),

name: 'Tom',

age: 28,

subject: 'Youtube'

}

]

students> db.students.find({subject:{$eq:"Maths"}})

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 20,

subject: 'Maths'

}

]

students> db.students.updateOne({name:"Tom"},{$set: {age:23}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

students> db.students.find()

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 20,

subject: 'Maths'

},

{

\_id: ObjectId("64de03f3a17b4ed899070121"),

name: 'Mody',

age: 18,

subject: 'Physics'

},

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology'

},

{

\_id: ObjectId("64de0453a17b4ed899070124"),

name: 'Tom',

age: 23,

subject: 'Youtube'

}

]

students> db.students.insertMany({},{grade:" "})

MongoInvalidArgumentError: Argument "docs" must be an array of documents

students> db.students.updateMany({},{grade:" "})

MongoInvalidArgumentError: Update document requires atomic operators

students> db.students.updateMany({},{grade:"A"})

MongoInvalidArgumentError: Update document requires atomic operators

students> db.students.updateOne({name:"Tom"},{grade:"B"})

MongoInvalidArgumentError: Update document requires atomic operators

students> db.students.updateOne({name:"Tom"},{$set:{grade:"B"}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

students> db.students.find()

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 20,

subject: 'Maths',

grade: 'A'

},

{

\_id: ObjectId("64de03f3a17b4ed899070121"),

name: 'Mody',

age: 18,

subject: 'Physics',

grade: 'B'

},

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry',

grade: 'C'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology',

grade: 'A'

},

{

\_id: ObjectId("64de0453a17b4ed899070124"),

name: 'Tom',

age: 23,

subject: 'Youtube',

grade: 'B'

}

]

students> db.students.deleteOne({name:"Mody"})

{ acknowledged: true, deletedCount: 1 }

students> db.students.find()

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 20,

subject: 'Maths',

grade: 'A'

},

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry',

grade: 'C'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology',

grade: 'A'

},

{

\_id: ObjectId("64de0453a17b4ed899070124"),

name: 'Tom',

age: 23,

subject: 'Youtube',

grade: 'B'

}

]

students> db.students.updateOne({name:"Chris"},{$set:{age:17}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

students> db.students.find()

[

{

\_id: ObjectId("64de02cba17b4ed899070120"),

name: 'Chris',

age: 17,

subject: 'Maths',

grade: 'A'

},

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry',

grade: 'C'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology',

grade: 'A'

},

{

\_id: ObjectId("64de0453a17b4ed899070124"),

name: 'Tom',

age: 23,

subject: 'Youtube',

grade: 'B'

}

]

students> db.students.deleteMany({age:{$lt:18}})

{ acknowledged: true, deletedCount: 1 }

students> db.students.find()

[

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry',

grade: 'C'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology',

grade: 'A'

},

{

\_id: ObjectId("64de0453a17b4ed899070124"),

name: 'Tom',

age: 23,

subject: 'Youtube',

grade: 'B'

}

]

students> db.students.createIndex("new\_index")

MongoServerError: Error in specification { name: "0\_n\_1\_e\_2\_w\_3\_\_\_4\_i\_5\_n\_6\_d\_7\_e\_8\_x", key: { 0: "n", 1: "e", 2: "w", 3: "\_", 4: "i", 5: "n", 6: "d", 7: "e", 8: "x" } } :: caused by :: Unknown index plugin 'x'

students> db.students.createIndex()

MongoshInvalidInputError: [COMMON-10001] Missing required argument at position 0 (Collection.createIndex)

students> db.students.getIndexes()

[

{ v: 2, key: { \_id: 1 }, name: '\_id\_' },

{ v: 2, key: { name: 1 }, name: 'name\_1' }

]

students> db.students.getIndexes()

[ { v: 2, key: { \_id: 1 }, name: '\_id\_' } ]

students> db.students.createIndex({name: 1})

name\_1

students> db.students.getIndexes()

[

{ v: 2, key: { \_id: 1 }, name: '\_id\_' },

{ v: 2, key: { name: 1 }, name: 'name\_1' }

]

students> db.students.find({name:"Layla"})

[

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology',

grade: 'A'

}

]

students> dbs.students.aggregate([$group:{\_id: null,averageAge:{$avg: "$age"}}])

Uncaught:

SyntaxError: Unexpected token, expected "," (1:30)

> 1 | dbs.students.aggregate([$group:{\_id: null,averageAge:{$avg: "$age"}}])

| ^

2 |

students> dbs.students.aggregate([{$group:{\_id: null,averageAge:{$avg: "$age"}}])

Uncaught:

SyntaxError: Unexpected token, expected "," (1:69)

> 1 | dbs.students.aggregate([{$group:{\_id: null,averageAge:{$avg: "$age"}}])

|

^

2 |

students> dbs.students.aggregate([{$group:{\_id: null,averageAge:{$avg: "$age"}}}])

ReferenceError: dbs is not defined

students> dbs.students.aggregate([{$group:{\_id: null,averageAge:{$avg: "$age"}}}])

ReferenceError: dbs is not defined

students> db.students.aggregate([

d: null,

... {

... $group: {

... \_i averageAge: { $avg: "$age" }

Uncaught:

SyntaxError: Unexpected token, expected "," (4:14)

2 | {

3 | $group: {

> 4 | \_i averageAge: { $avg: "$age" }

| ^

5 |

students> }

Uncaught:

SyntaxError: Unexpected token (1:4)

> 1 | }

| ^

2 |

students> }

Uncaught:

SyntaxError: Unexpected token (1:2)

> 1 | }

| ^

2 |

students> ])

Uncaught:

SyntaxError: Unexpected token (1:0)

> 1 | ])

| ^

2 |

students> db.students.aggregate([{$group: {\_id: null,averageAge: { $avg: "$age"}}}])

[ { \_id: null, averageAge: 23.333333333333332 } ]

students> db.students.aggregate([{$group: {\_id: "$subject",averageAge: { $sum: 1}}}])

[

{ \_id: 'chemistry', averageAge: 1 },

{ \_id: 'Geology', averageAge: 1 },

{ \_id: 'Youtube', averageAge: 1 }

]

students> db.createCollection("restaurants")

{ ok: 1 }

students> db.restaurants.insertOne({

cuisine: "Italian",

... name: "Tasty Eats",

... address: "123 Main St",

... grades: [

... { grade: "A", score: 95 },

... { grade: "B", score: 82 },

... { grade: "A", score: 98 }

... ]

... })

{

acknowledged: true,

insertedId: ObjectId("64de09e4a17b4ed899070125")

}

students> db.students.find()

[

{

\_id: ObjectId("64de040da17b4ed899070122"),

name: 'Gary',

age: 22,

subject: 'chemistry',

grade: 'C'

},

{

\_id: ObjectId("64de0424a17b4ed899070123"),

name: 'Layla',

age: 25,

subject: 'Geology',

grade: 'A'

},

{

\_id: ObjectId("64de0453a17b4ed899070124"),

name: 'Tom',

age: 23,

subject: 'Youtube',

grade: 'B'

}

]

students> show dbs

Library 184.00 KiB

admin 40.00 KiB

config 120.00 KiB

local 72.00 KiB

school 8.00 KiB

students 100.00 KiB

students> show collections

restaurants

students

students> use restraunts

switched to db restraunts

restraunts> db.restraunts.find()

restraunts> db.restraunts.find()

restraunts> db.restraunts.find({name: "Tasty Eats"})

restraunts> db.restraunts.find({name: "Tasty Eats"})

restraunts> use students

switched to db students

students> db.restraunts.find()

students> use restaurants

switched to db restaurants

restaurants> db.restaurants.insertOne({

cuisine: "Italian",

... name: "Tasty Eats",

... address: "123 Main St",

e: "B", score: 82 },

... grades: [

... { grade: "A", score: 95 },

... { grad { grade: "A", score: 98 }

Uncaught:

SyntaxError: Unexpected token, expected "," (6:14)

4 | grades: [

5 | { grade: "A", score: 95 },

> 6 | { grad { grade: "A", score: 98 }

| ^

7 |

restaurants> ]

Uncaught:

SyntaxError: Unexpected token (1:2)

> 1 | ]

| ^

2 |

restaurants> })

Uncaught:

SyntaxError: Unexpected token (1:0)

> 1 | })

| ^

2 |

restaurants>

restaurants> db.restaurants.insertOne({ name: "Tasty Eats", address: "123

Main St", grades: [ { grade: "A", score: 95 }, { grad { grade: "A", score: 98 }

Uncaught:

SyntaxError: Unexpected token, expected "," (1:118)

> 1 | db.restaurants.insertOne({ name: "Tasty Eats", address: "123 Main St", grades: [ { grade: "A", score: 95 }, { grad { grade: "A", score: 98

}

|

^

2 |

restaurants> db.restaurants.insertOne({ name: "Tasty Eats", address: "123

Main St", grades: [ { grade: "A", score: 95 },{grade:"B", score: 82},{ grade: "A", score: 98 }

...

...

restaurants> db.restaurants.insertOne({ name: "Tasty Eats", address: "123

Main St", grades: [ { grade: "A", score: 95 },{grade:"B", score: 82},{ grade: "A", score: 98 }})

Uncaught:

SyntaxError: Unexpected token, expected "," (1:155)

> 1 | db.restaurants.insertOne({ name: "Tasty Eats", address: "123 Main St", grades: [ { grade: "A", score: 95 },{grade:"B", score: 82},{ grade: "A", score: 98 }})

|

^

2 |

restaurants> db.restaurants.insertOne({

cuisine: "Italian",

... name: "Tasty Eats",

... address: "123 Main St",

... grades: [

... { grade: "A", score: 95 },

... { grade: "B", score: 82 },

... { grade: "A", score: 98 }

...

... ]

... })

{

acknowledged: true,

insertedId: ObjectId("64de0bc8a17b4ed899070126")

}

restaurants> db.restaurants.find()

[

{

\_id: ObjectId("64de0bc8a17b4ed899070126"),

name: 'Tasty Eats',

address: '123 Main St',

grades: [

{ grade: 'A', score: 95 },

{ grade: 'B', score: 82 },

{ grade: 'A', score: 98 }

]

}

]

restaurants> db.restaurants.find({ "grades.grade": "A" })

[

{

\_id: ObjectId("64de0bc8a17b4ed899070126"),

name: 'Tasty Eats',

address: '123 Main St',

grades: [

{ grade: 'A', score: 95 },

{ grade: 'B', score: 82 },

{ grade: 'A', score: 98 }

]

}

]

restaurants>