Microsoft Windows [Version 10.0.22631.4169]

(c) Microsoft Corporation. All rights reserved.

C:\Users\Purvi>mongosh

Current Mongosh Log ID: 66f1b4a369ae0fd330117b7a

Connecting to:
mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.2.4

Using MongoDB: 7.0.8

Using Mongosh: 2.2.4

mongosh 2.3.0 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

2024-09-12T17:55:46.422+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

---

test> use UniversityDB

switched to db UniversityDB

UniversityDB> db.students.insertMany([

... { name: "Alice", major: "Computer Science", gpa: 3.9, email: "alice@example.com" },

... { name: "Bob", major: "Mathematics", gpa: 3.5, email: "bob@example.com" },

... { name: "Charlie", major: "Computer Science", gpa: 3.7, email: "charlie@example.com" },

... { name: "David", major: "Mathematics", gpa: 3.2, email: "david@example.com" },

... { name: "Eve", major: "Physics", gpa: 3.8, email: "eve@example.com" }

...])

{

acknowledged: true,

insertedIds: {

```
    '0': ObjectId('66f1b64069ae0fd330117b7b'),
    '1': ObjectId('66f1b64069ae0fd330117b7c'),
    '2': ObjectId('66f1b64069ae0fd330117b7d'),
    '3': ObjectId('66f1b64069ae0fd330117b7e'),
    '4': ObjectId('66f1b64069ae0fd330117b7f')
  }
}
UniversityDB> db.students.find().pretty()
[
  {
    _id: ObjectId('66f1b64069ae0fd330117b7b'),
    name: 'Alice',
    major: 'Computer Science',
    gpa: 3.9,
    email: 'alice@example.com'
  },
  {
    _id: ObjectId('66f1b64069ae0fd330117b7c'),
    name: 'Bob',
    major: 'Mathematics',
    gpa: 3.5,
    email: 'bob@example.com'
  },
  {
    _id: ObjectId('66f1b64069ae0fd330117b7d'),
    name: 'Charlie',
    major: 'Computer Science',
    gpa: 3.7,
    email: 'charlie@example.com'
  },
  {
```

```
    _id: ObjectId('66f1b64069ae0fd330117b7e'),
    name: 'David',
    major: 'Mathematics',
    gpa: 3.2,
    email: 'david@example.com'
  },
  {
    _id: ObjectId('66f1b64069ae0fd330117b7f'),
    name: 'Eve',
    major: 'Physics',
    gpa: 3.8,
    email: 'eve@example.com'
  }
]
UniversityDB> db.students.createIndex({field:1})
field_1
UniversityDB> db.students.dropIndex({field:1})
{ nIndexesWas: 2, ok: 1 }
UniversityDB> db.students.getIndexes()
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
UniversityDB> db.students.createIndex({name:1})
name_1
UniversityDB> db.students.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' }
]
UniversityDB> db.students.createIndex({major:1,gpa:-1})
major_1_gpa_-1
UniversityDB> db.students.getIndexes()
[
```

```
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' },
  { v: 2, key: { major: 1, gpa: -1 }, name: 'major_1_gpa_-1' }
]
UniversityDB> db.student.createIndexes({email:1},{unique:true})
TypeError: e.map is not a function
UniversityDB> db.student.createIndex({email:1},{unique:true})
email_1
UniversityDB> db.students.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' },
  { v: 2, key: { major: 1, gpa: -1 }, name: 'major_1_gpa_-1' }
]
UniversityDB> db.students.dropIndex({email:1})
MongoshInternalError[IndexNotFound]: can't find index with key: { email: 1 }
UniversityDB> db.students.createIndex({email:1},{unique:true})
email_1
UniversityDB> db.students.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' },
  { v: 2, key: { major: 1, gpa: -1 }, name: 'major_1_gpa_-1' },
  { v: 2, key: { email: 1 }, name: 'email_1', unique: true }
]
UniversityDB> db.students.dropIndex({email:1})
{ nIndexesWas: 4, ok: 1 }
UniversityDB> db.students.dropIndex({major:1,gpa:-1})
{ nIndexesWas: 3, ok: 1 }
UniversityDB> db.students.getIndexes()
[
```

```
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'name_1' }
]
UniversityDB> db.students.stats().indexSizes
{ _id_: 20480, name_1: 20480 }
UniversityDB>
```