Name : Aditya Tukram Kamble Roll No : L011

Practical 4 : INDEXING USING Mangodb

Date:-17/01/2024 Submission Date:- 24/01/2024

Write-up: -

· Indexing in mongodb

· Types of indexing

· Document document-oriented NoSQL

1. Mongo DB indexing

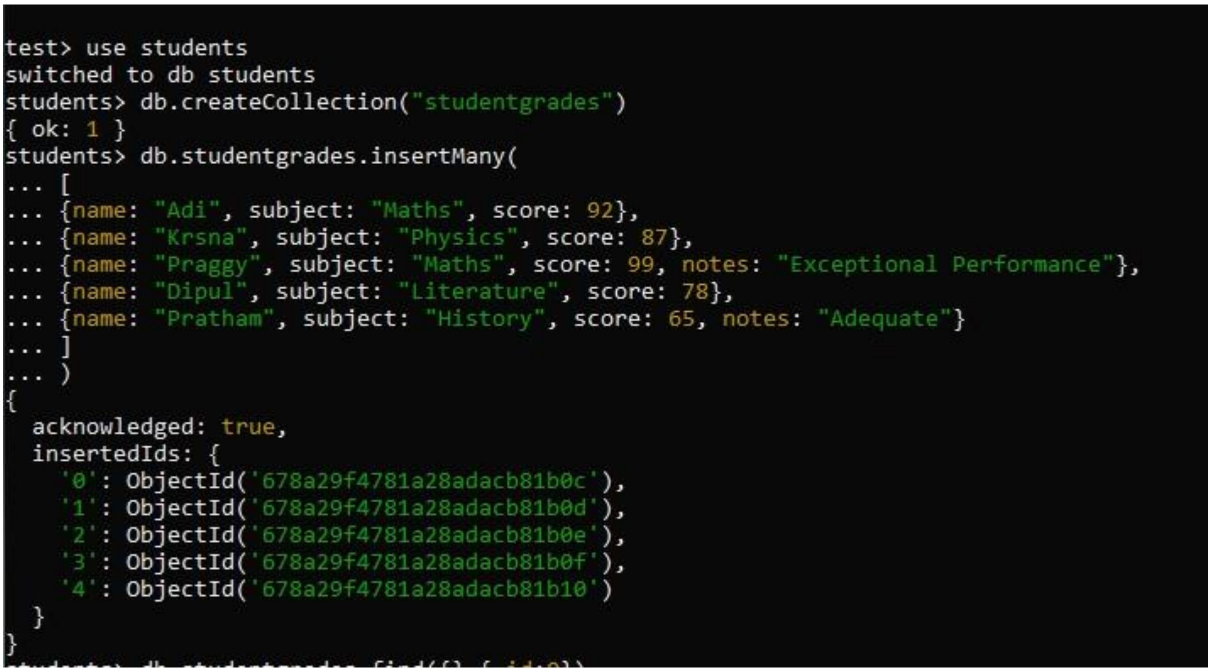
a. Create index in Mongo DB

b. Finding the indexes in a collection

c. Drop indexes in a collection

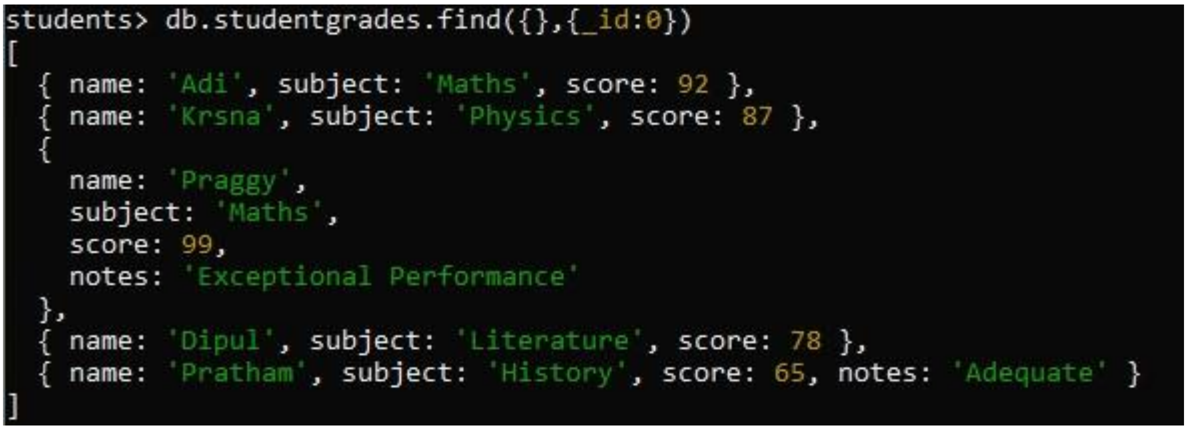
d. Drop all the indexes use students db.createCollection("studentgrades")

```
   db.studentgrades.insertMany(
[
{name: "Adi", subject: "Maths", score: 92},
{name: "Krsna", subject: "Physics", score: 87},
{name: "Praggy", subject: "Maths", score: 99, notes: "Exceptional Performance"},
{name: "Dipul", subject: "Literature", score: 78},
{name: "Pratham", subject: "History", score: 65, notes: "Adequate"}
]
)
```



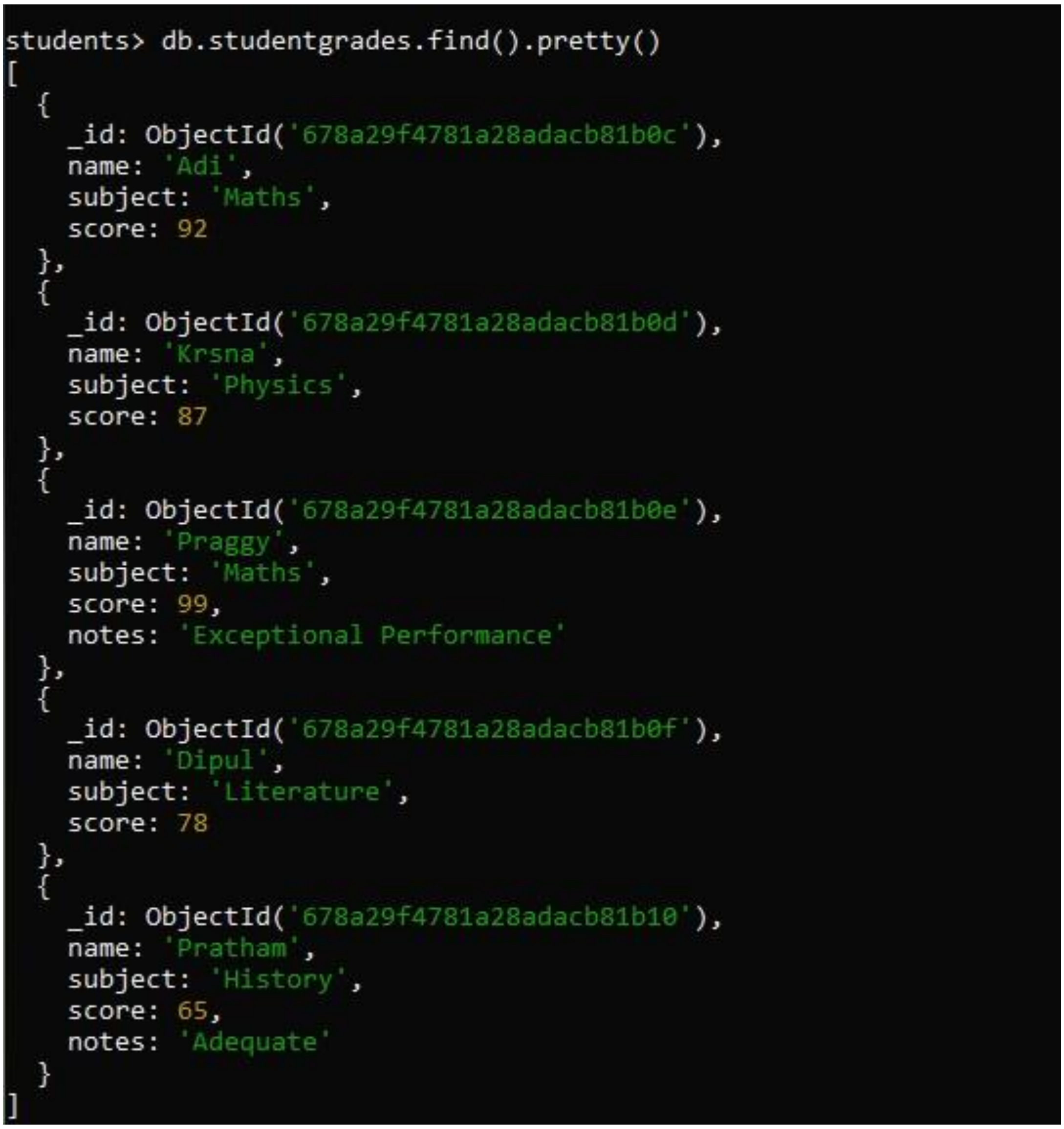
```
test> use students
switched to db students
students> db.createCollection("studentgrades")
{ ok: 1 }
students> db.studentgrades.insertMany(
... [
... {name: "Adi", subject: "Maths", score: 92},
... {name: "Krsna", subject: "Physics", score: 87},
... {name: "Praggy", subject: "Maths", score: 99, notes: "Exceptional Performance"},
... {name: "Dipul", subject: "Literature", score: 78},
... {name: "Pratham", subject: "History", score: 65, notes: "Adequate"}
... ]
... )
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('678a29f4781a28adacb81b0c'),
    '1': ObjectId('678a29f4781a28adacb81b0d'),
    '2': ObjectId('678a29f4781a28adacb81b0e'),
    '3': ObjectId('678a29f4781a28adacb81b0f'),
    '4': ObjectId('678a29f4781a28adacb81b10')
  }
}
students> db.studentgrades.find({},{_id:0})
```

db.studentgrades.find({},{_id:0})



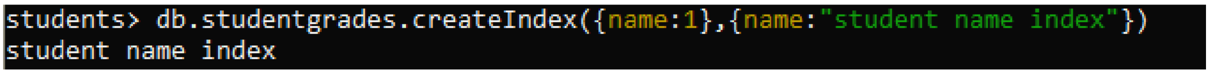
```
students> db.studentgrades.find({},{_id:0})
[
  { name: 'Adi', subject: 'Maths', score: 92 },
  { name: 'Krsna', subject: 'Physics', score: 87 },
  {
    name: 'Praggy',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  { name: 'Dipul', subject: 'Literature', score: 78 },
  { name: 'Pratham', subject: 'History', score: 65, notes: 'Adequate' }
]
```

db.studentgrades.find().pretty()



```
students> db.studentgrades.find().pretty()
[
  {
    _id: ObjectId('678a29f4781a28adacb81b0c'),
    name: 'Adi',
    subject: 'Maths',
    score: 92
  },
  {
    _id: ObjectId('678a29f4781a28adacb81b0d'),
    name: 'Krsna',
    subject: 'Physics',
    score: 87
  },
  {
    _id: ObjectId('678a29f4781a28adacb81b0e'),
    name: 'Praggy',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  {
    _id: ObjectId('678a29f4781a28adacb81b0f'),
    name: 'Dipul',
    subject: 'Literature',
    score: 78
  },
  {
    _id: ObjectId('678a29f4781a28adacb81b10'),
    name: 'Pratham',
    subject: 'History',
    score: 65,
    notes: 'Adequate'
  }
]
```

db.studentgrades.createIndex({name: 1}, {name: "student name index"})

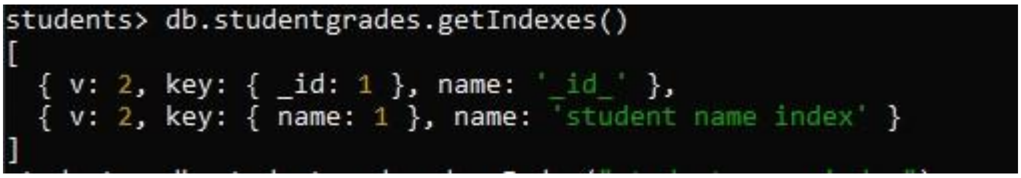


```
students> db.studentgrades.createIndex({name:1},{name:"student name index"})
student name index
```

Finding indexes You can find all the available indexes in a MongoDB collection by using the getIndexes method. This will return all the indexes in a specific collection. db..getIndexes()

Let’s view all the indexes in the studentgrades collection using the following command:

db.studentgrades.getIndexes()



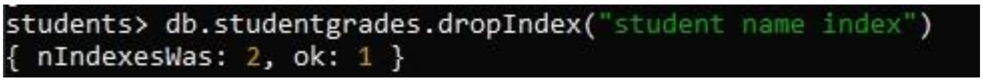
```
students> db.studentgrades.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'student name index' }
]
```

Dropping indexes To delete an index from a collection, use the dropIndex method while specifying the index name to be dropped. db..dropIndex() Let’s remove the user-created index with the index name student name index, as shown below.

db.studentgrades.dropIndex("student name index")

You can also use the index field value for removing an index without a defined name:
db.studentgrades.dropIndex({name:1})

The dropIndexes command can also drop all the indexes excluding the default _id index.
db.studentgrades.dropIndexes()

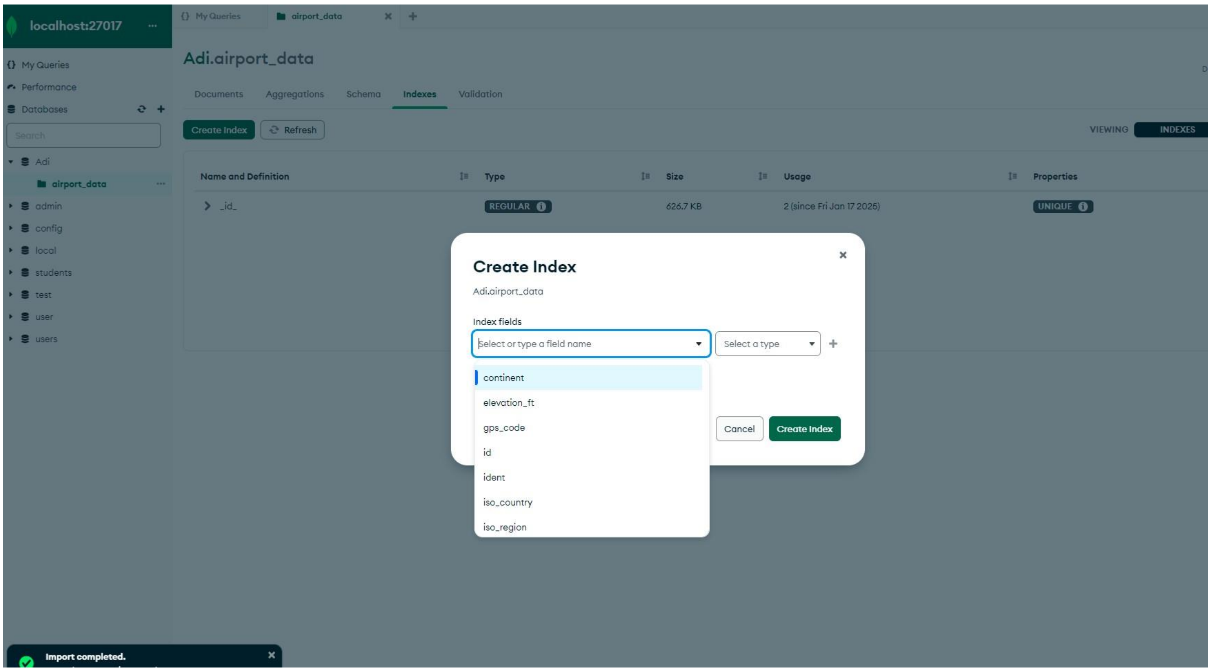


```
students> db.studentgrades.dropIndex("student name index")
{ nIndexesWas: 2, ok: 1 }
```

2. Create all the types of indexes (discussed in class) which will help in finding certain words in a document by using AIRPORT(dataset)

Ans. step 1 : import data and create index

Name : Aditya Tukram Kamble   Roll No : L011



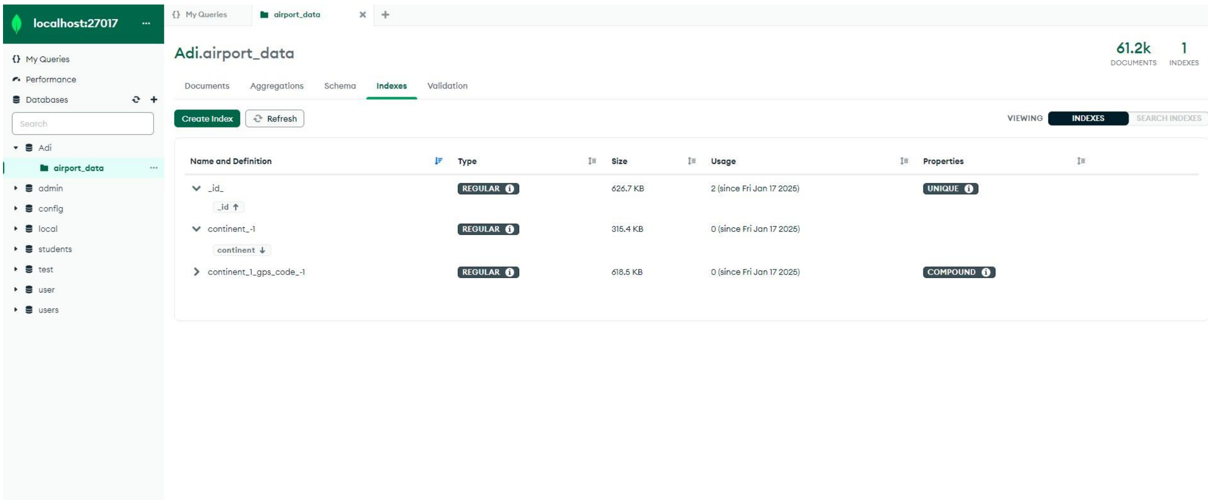
Name and Definition	Type	Size	Usage	Properties
_id_	REGULAR	626.7 KB	2 (since Fri Jan 17 2025)	UNIQUE

Create Index

Adi.airport_data

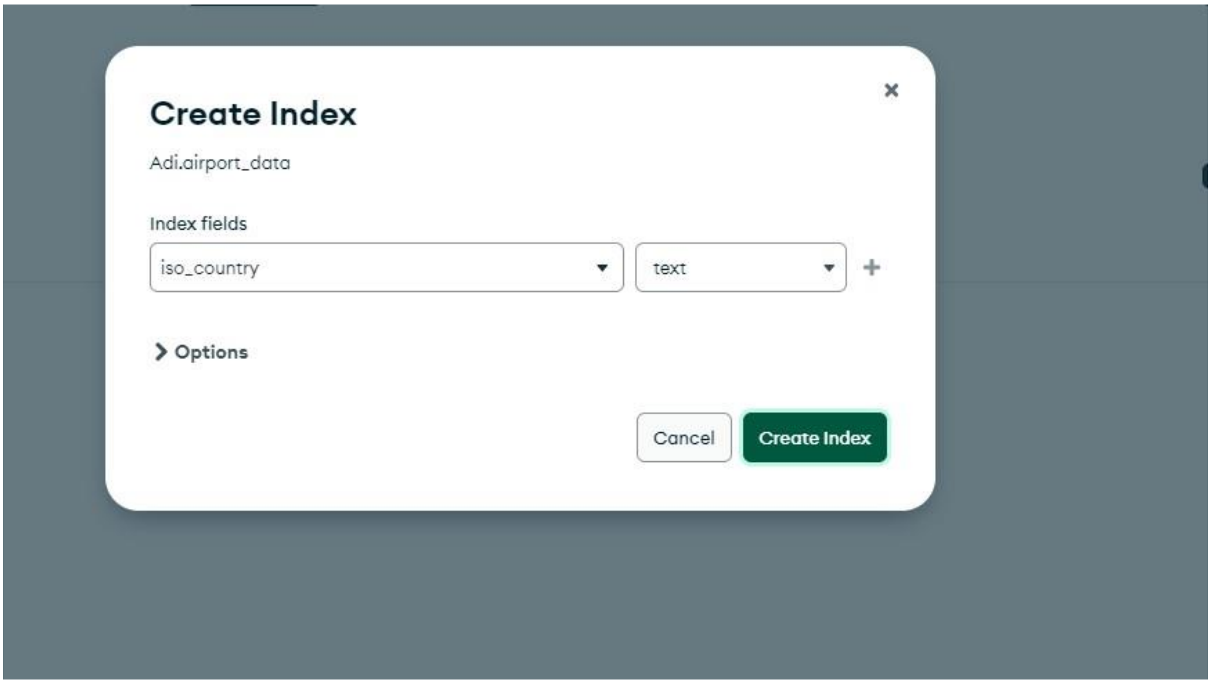
Index fields

Import completed.



Name and Definition	Type	Size	Usage	Properties
_id_ _id ↑	REGULAR	626.7 KB	2 (since Fri Jan 17 2025)	UNIQUE
continent_-1 continent ↓	REGULAR	315.4 KB	0 (since Fri Jan 17 2025)	
continent_1_gps_code_-1	REGULAR	618.5 KB	0 (since Fri Jan 17 2025)	COMPOUND

Text indexing

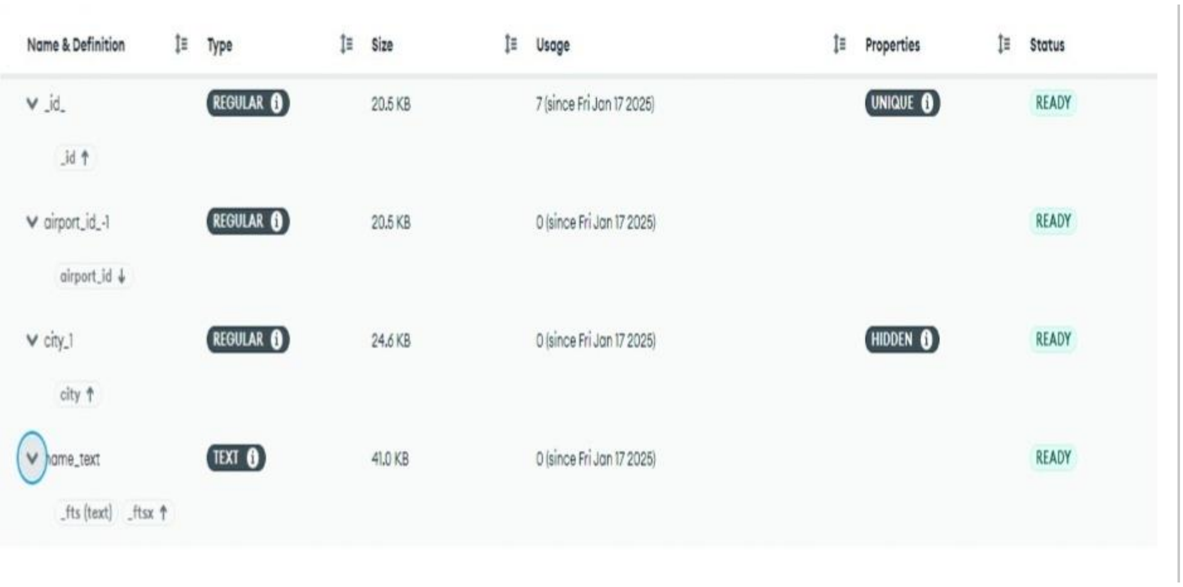


Create Index

Adi.airport_data

Index fields

iso_country ▾ text ▾



Name & Definition	Type	Size	Usage	Properties	Status
_id_ _id ↑	REGULAR	20.5 KB	7 (since Fri Jan 17 2025)	UNIQUE	READY
airport_id_-1 airport_id ↓	REGULAR	20.5 KB	0 (since Fri Jan 17 2025)		READY
city_1 city ↑	REGULAR	24.6 KB	0 (since Fri Jan 17 2025)	HIDDEN	READY
name_text _fts (text) _ftsx ↑	TEXT	41.0 KB	0 (since Fri Jan 17 2025)		READY