

## Practical-4: INDEXING USING Mangodb

Class:MSC DSAI

Roll No:L-09

```
test> use students
switched to db students
students> db.createCollection("studentgrades")
{ ok: 1 }
students> db.studentgrades.insertMany(
... [
... {name: "Barry", subject: "Maths", score: 92},
... {name: "Kent", subject: "Physics", score: 87},
... {name: "Harry", subject: "Maths", score: 99, notes: "Exceptional Performance"},
... {name: "Alex", subject: "Literature", score: 78},
... {name: "Tom", subject: "History", score: 65, notes: "Adequate"}
... ]
... )
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('678a26ff09f5e0ff63cb0ce2'),
    '1': ObjectId('678a26ff09f5e0ff63cb0ce3'),
    '2': ObjectId('678a26ff09f5e0ff63cb0ce4'),
    '3': ObjectId('678a26ff09f5e0ff63cb0ce5'),
    '4': ObjectId('678a26ff09f5e0ff63cb0ce6')
  }
}
```

```

students> db.studentgrades.find({}, {_id:0})
[
  { name: 'Barry', subject: 'Maths', score: 92 },
  { name: 'Kent', subject: 'Physics', score: 87 },
  {
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  { name: 'Alex', subject: 'Literature', score: 78 },
  { name: 'Tom', subject: 'History', score: 65, notes: 'Adequate' }
]
students> db.studentgrades.find().pretty()
[
  {
    _id: ObjectId('678a26ff09f5e0ff63cb0ce2'),
    name: 'Barry',
    subject: 'Maths',
    score: 92
  },
  {
    _id: ObjectId('678a26ff09f5e0ff63cb0ce3'),
    name: 'Kent',
    subject: 'Physics',
    score: 87
  },
  {
    _id: ObjectId('678a26ff09f5e0ff63cb0ce4'),
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  {
    _id: ObjectId('678a26ff09f5e0ff63cb0ce5'),
    name: 'Alex',
    subject: 'Literature',
    score: 78
  },
  {
    _id: ObjectId('678a26ff09f5e0ff63cb0ce6'),
    name: 'Tom',
    subject: 'History',
    score: 65,
    notes: 'Adequate'
  }
]

```

```

  {
    _id: ObjectId('678a26ff09f5e0ff63cb0ce4'),
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  {
    _id: ObjectId('678a26ff09f5e0ff63cb0ce5'),
    name: 'Alex',
    subject: 'Literature',
    score: 78
  },
  {
    _id: ObjectId('678a26ff09f5e0ff63cb0ce6'),
    name: 'Tom',
    subject: 'History',
    score: 65,
    notes: 'Adequate'
  }
]

```

```


students> db.studentgrades.createIndex( {name: 1}, {name: "student name index"} )
student name index
students> db.studentgrades.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_1' },
  { v: 2, key: { name: 1 }, name: 'student name index' }
]
students> db.studentgrades.dropIndex("student name index")
{ nIndexesWas: 2, ok: 1 }
students> db.studentgrades.dropIndex({name:1})
MongoServerError[IndexNotFound]: can't find index with key: { name: 1 }
students> db.studentgrades.createIndex( {name: 1} )
name_1
students> db.studentgrades.dropIndex({name:1})
{ nIndexesWas: 2, ok: 1 }
students> db.studentgrades.dropIndexes()
{
  nIndexesWas: 1,
  msg: 'non-_id indexes dropped for collection',
  ok: 1
}

```

## New Connection

Manage your connection settings

URI ⓘ

Edit Connection String 

mongodb://localhost:27017/

Name

mansi

Color

Teal

☐ Favorite this connection  
 Favoriting a connection will pin it to the top of your list of connections

> Advanced Connection Options

Cancel

Save

Connect

Save & Connect

### How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.

[See example](#)

### How do I format my connection string?

[See example](#)

Compass

My Queries

CONNECTIONS (1)

Search connections

mansi

admin

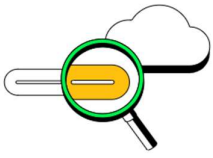
config

local

students

Welcome

+



Welcome to MongoDB Compass

To get started, connect to an existing server or

+ Add new connection

New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using [MongoDB Atlas](#)

CREATE FREE CLUSTER

Compass is ready to update to 1.46.1!

RESTART

Create Database

Database Name

Airport

Collection Name

Practical

☐ Time-Series

Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> Additional preferences

(e.g. Custom collation, Capped, Clustered collections)

Cancel

Create Database



### This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Import Data

## Import

To collection Airport,Practical

Import file: airports.csv

### Options

Select delimiter Comma

☒ Ignore empty strings

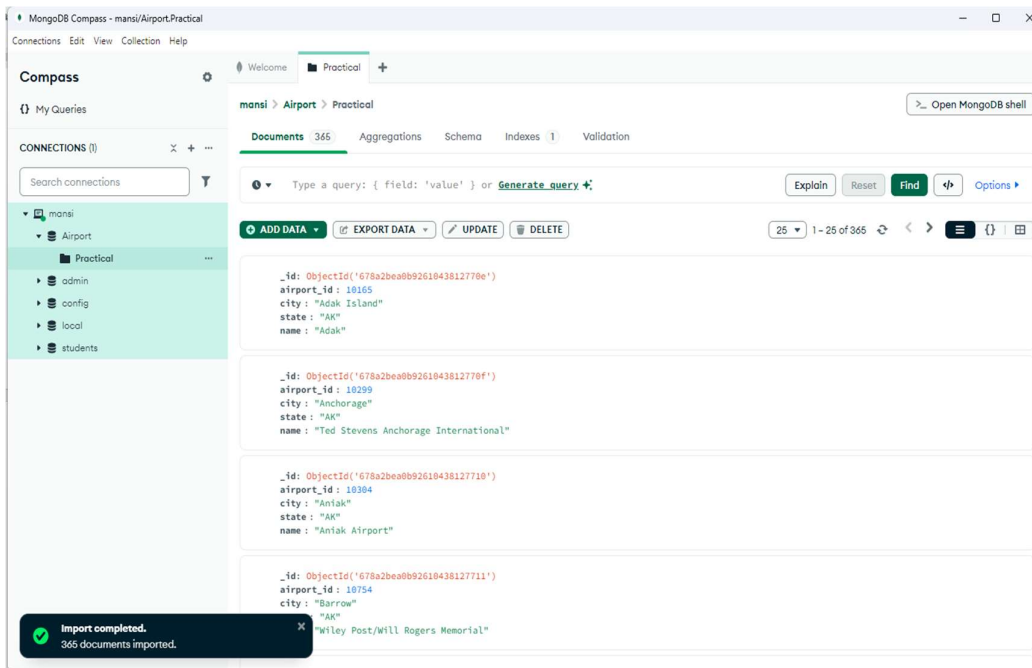
☐ Stop on errors

Specify Fields and Types [Learn more about data types](#)

|   | <input checked="" type="checkbox"/> airport_id<br>Int32 | <input checked="" type="checkbox"/> city<br>String | <input checked="" type="checkbox"/> state<br>String | <input checked="" type="checkbox"/> name<br>String |
|---|---|--|---|--|
| 1 | 10165   | Adak Island  | AK  | Adak   |
| 2 | 10299   | Anchorage  | AK  | Ted Stevens Anchorage International                |
| 3 | 10304   | Aniak  | AK  | Aniak Airport                                      |
| 4 | 10754   | Barrow   | AK  | Wiley Post/Will Rogers Memorial                    |
| 5 | 10551   | Bethel   | AK  | Bethel Airport                                     |

Cancel

Import



## Create Index

Airport.Practical

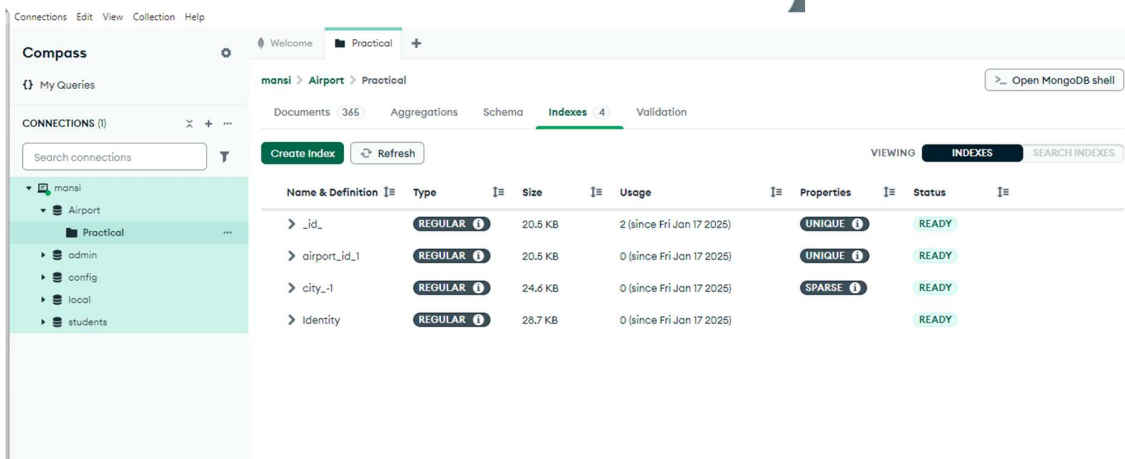
Index fields

airport\_id 1 (asc) +

> Options

Cancel

Create Index



## Create Index



Airport.Practical

Index fields

|       |           |   |   |
|-------|-----------|---|---|
| name  | 1 (asc)   | + | - |
| state | -1 (desc) | + | - |

### Options

☐ **Create unique index**

A unique index ensures that the indexed fields do not store duplicate values; i.e. enforces uniqueness for the indexed fields.

☒ **Index name**

Enter the name of the index to create, or leave blank to have MongoDB create a default name for the index.

Compound Optional

☐ **Create TTL**

TTL indexes are special single-field indexes that MongoDB can use to

Cancel

Create Index

> Compound

REGULAR ⓘ

28.7 KB

0 (since Fri Jan 17 2025)

COMPOUND ⓘ

READY

## Create Index



Airport.Practical

Index fields

state

text

+

### Options

☐ **Create unique index**

A unique index ensures that the indexed fields do not store duplicate values; i.e. enforces uniqueness for the indexed fields.

☐ **Index name**

Enter the name of the index to create, or leave blank to have MongoDB create a default name for the index.

☒ **Create TTL**

TTL indexes are special single-field indexes that MongoDB can use to automatically remove documents from a collection after a certain amount of time or at a specific clock time.

**seconds**

Cancel

Create Index

state\_text

TEXT

20.5 KB

0 (since Fri Jan 17 2025)

TTL

READY