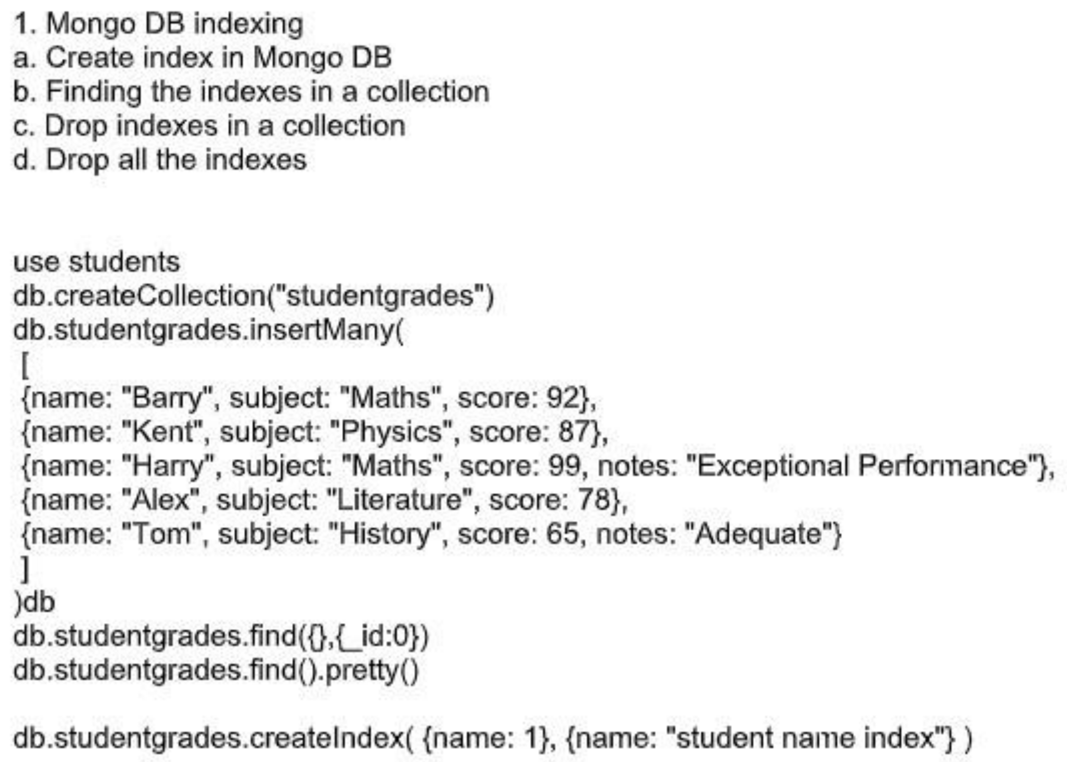
PRACTICAL N0 – 4

Aim: Indexing using Mongodb



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

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

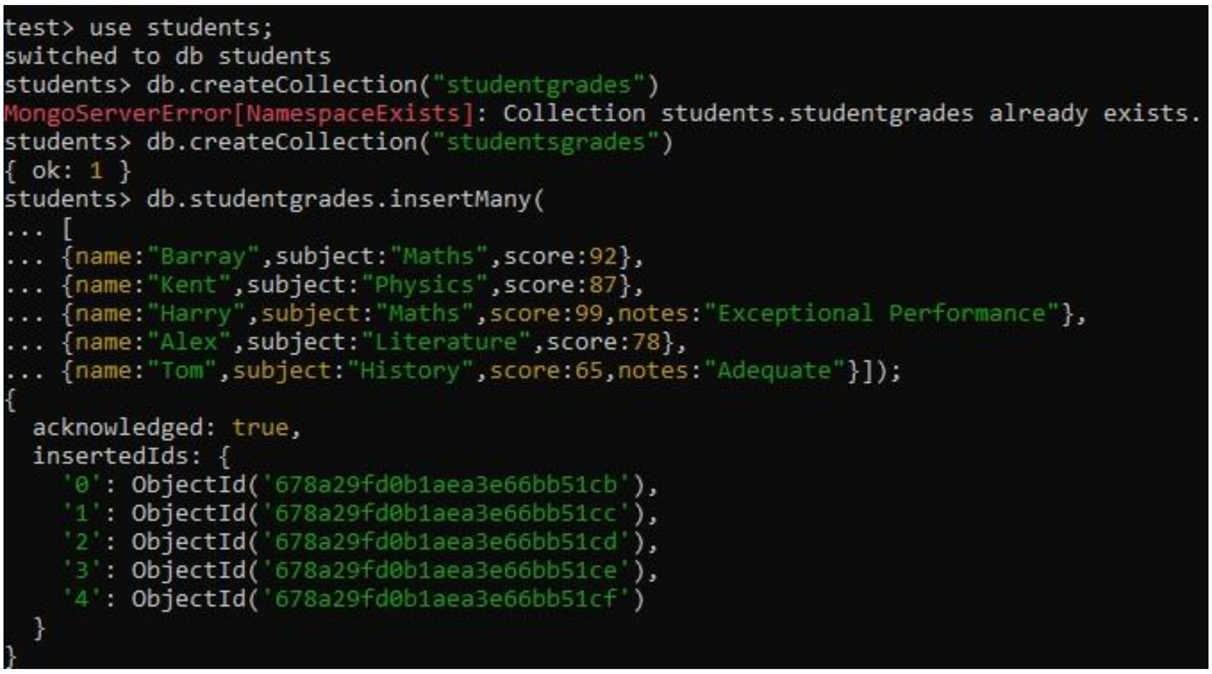
Code:

```
use students; db.createCollection("studentsgrades")

db.studentgrades.insertMany(

[

{name:"Barray",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"}]);

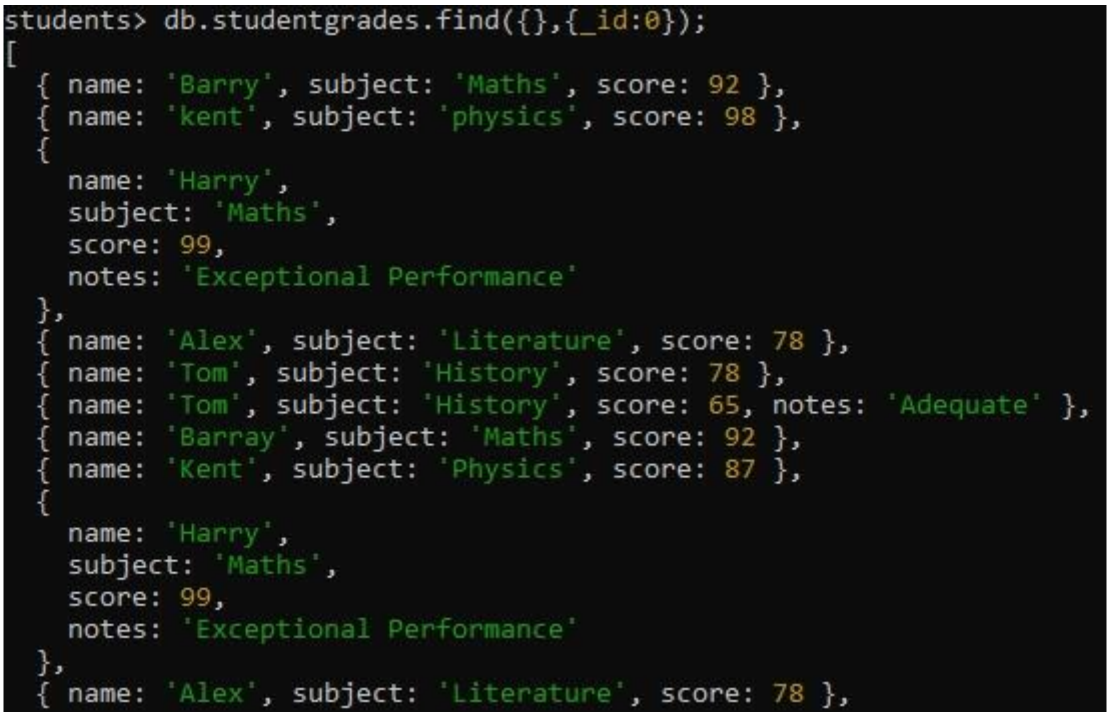
Output:



```
test> use students;
switched to db students
students> db.createCollection("studentgrades")
MongoServerError[NamespaceExists]: Collection students.studentgrades already exists.
students> db.createCollection("studentsgrades")
{ ok: 1 }
students> db.studentgrades.insertMany(
... [
... {name:"Barray",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('678a29fd0b1aea3e66bb51cb'),
    '1': ObjectId('678a29fd0b1aea3e66bb51cc'),
    '2': ObjectId('678a29fd0b1aea3e66bb51cd'),
    '3': ObjectId('678a29fd0b1aea3e66bb51ce'),
    '4': ObjectId('678a29fd0b1aea3e66bb51cf')
  }
}
```

Code:

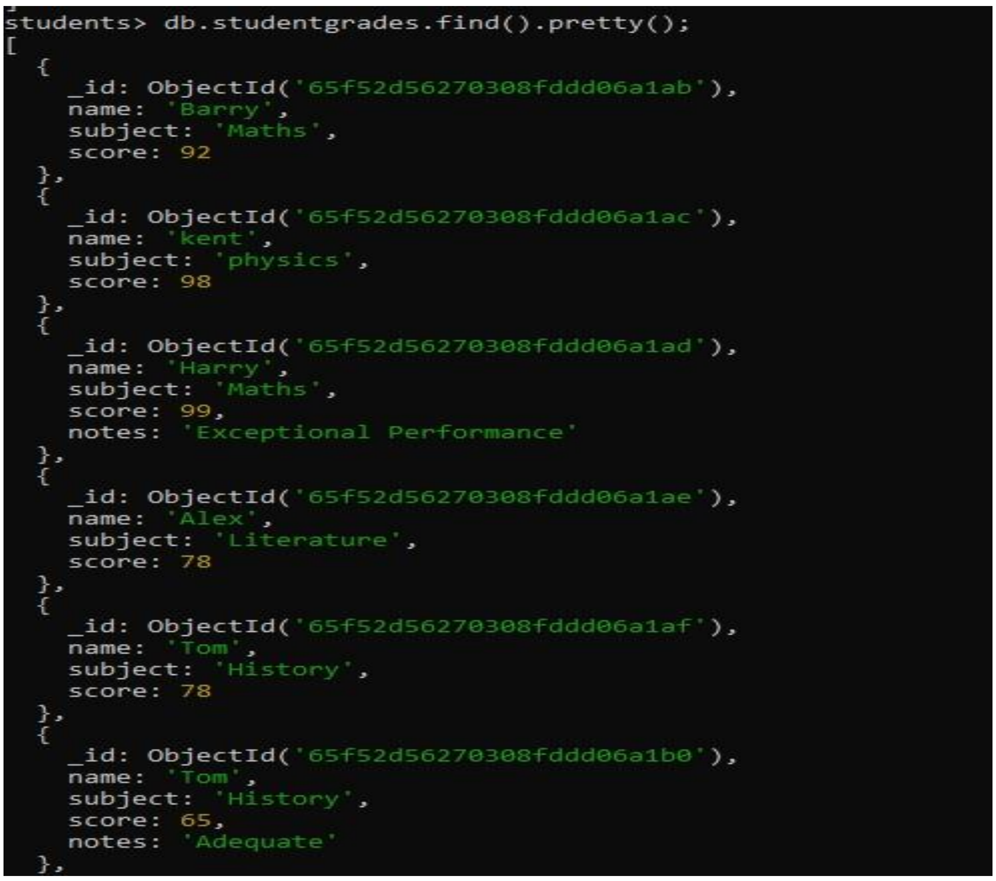
db.studentgrades.find({},{_id:0}); **Output:**



```
students> db.studentgrades.find({},{_id:0});
[
  { name: 'Barry', subject: 'Maths', score: 92 },
  { name: 'kent', subject: 'physics', score: 98 },
  {
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  { name: 'Alex', subject: 'Literature', score: 78 },
  { name: 'Tom', subject: 'History', score: 78 },
  { name: 'Tom', subject: 'History', score: 65, notes: 'Adequate' },
  { name: 'Barray', 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 },
```

Code:

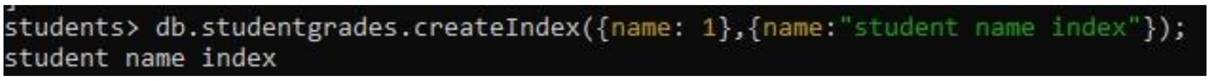
db.studentgrades.find().pretty(); **Output:**



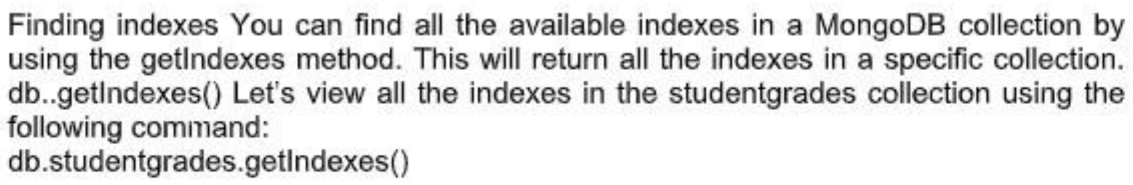
```
students> db.studentgrades.find().pretty();
[
  {
    _id: ObjectId('65f52d56270308fddd06a1ab'),
    name: 'Barry',
    subject: 'Maths',
    score: 92
  },
  {
    _id: ObjectId('65f52d56270308fddd06a1ac'),
    name: 'kent',
    subject: 'physics',
    score: 98
  },
  {
    _id: ObjectId('65f52d56270308fddd06a1ad'),
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  {
    _id: ObjectId('65f52d56270308fddd06a1ae'),
    name: 'Alex',
    subject: 'Literature',
    score: 78
  },
  {
    _id: ObjectId('65f52d56270308fddd06a1af'),
    name: 'Tom',
    subject: 'History',
    score: 78
  },
  {
    _id: ObjectId('65f52d56270308fddd06a1b0'),
    name: 'Tom',
    subject: 'History',
    score: 65,
    notes: 'Adequate'
  },
```

Code:

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



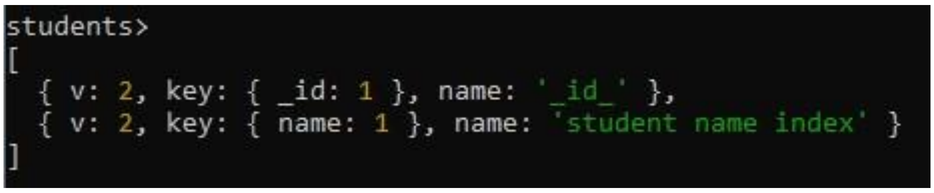
```
students> db.studentgrades.createIndex({name: 1},{name:"student name index"});
student name index
```



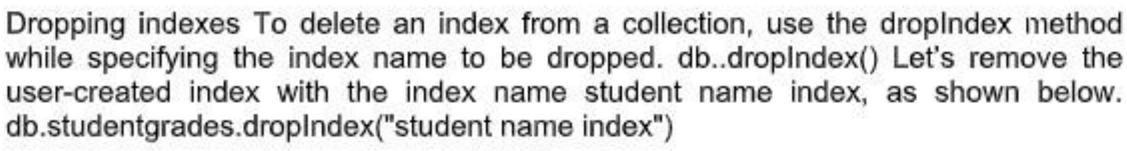
Finding indexes You can find all the available indexes in a MongoDB collection by using the getIndexes method. This will return all the indexes in a specific collection. db..getIndexes() Let's view all the indexes in the studentgrades collection using the following command:
db.studentgrades.getIndexes()

Code:

db.studentgrades.getIndexes(); **Output:**



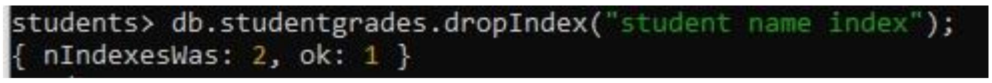
```
students>
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'student name index' }
]
```



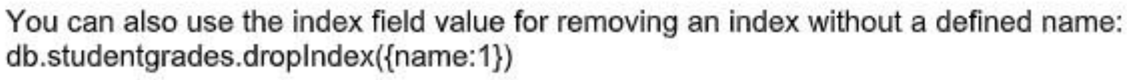
Dropping indexes To delete an index from a collection, use the dropIndex method while specifying the index name to be dropped. db..dropIndex() Let's remove the user-created index with the index name student name index, as shown below. db.studentgrades.dropIndex("student name index")

Code:

db.studentgrades.dropIndex("student name index"); **Output:**



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

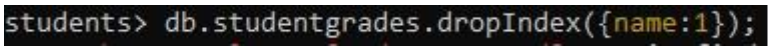


You can also use the index field value for removing an index without a defined name: db.studentgrades.dropIndex({name:1})

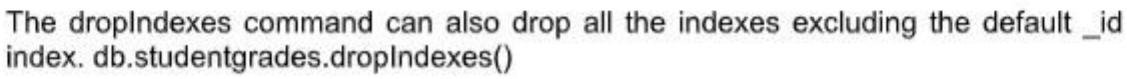
Code:

db.studentgrades.dropIndex({name:1});

Output:



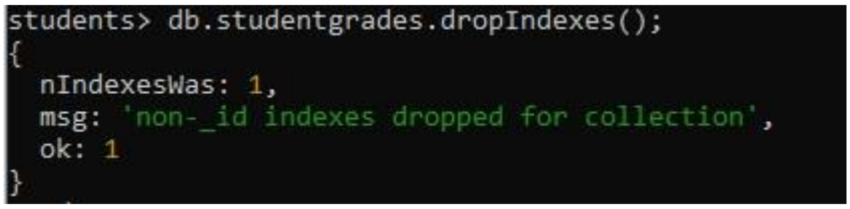
```
students> db.studentgrades.dropIndex({name:1});
```



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

Code:

db.studentgrades.dropIndexes(); **Output:**

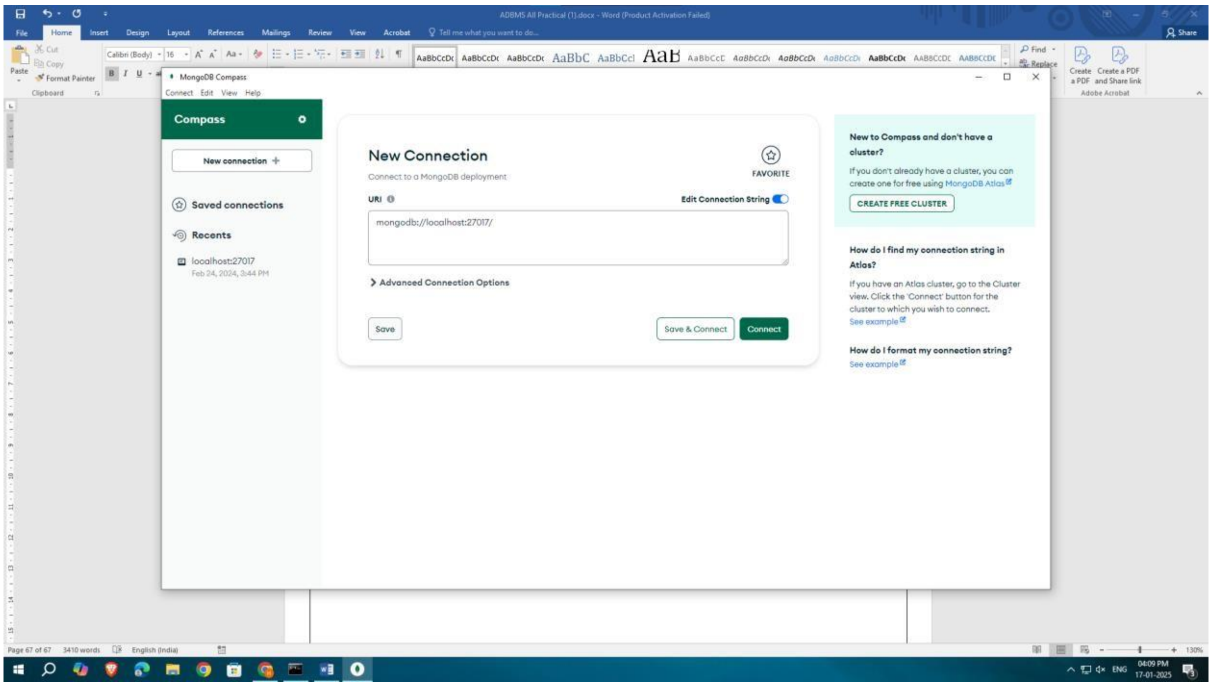


```
students> db.studentgrades.dropIndexes();
{
  nIndexesWas: 1,
  msg: 'non-_id indexes dropped for collection',
  ok: 1
}
```

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

Step 1: Go to mongodb compass

Step 2: Connect to the localhost



New Connection

Connect to a MongoDB deployment

URI: mongodb://localhost:27017/

New to Compass and don't have a cluster?

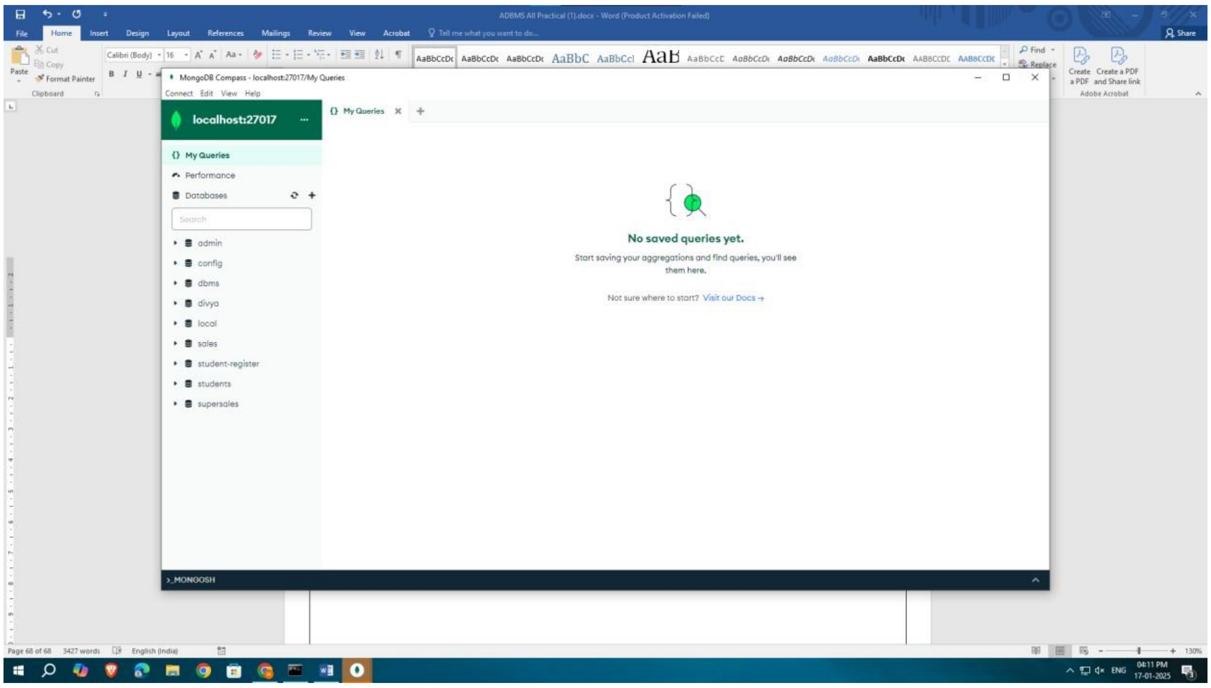
If you don't already have a cluster, you can create one for free using MongoDB Atlas

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.

How do I format my connection string?

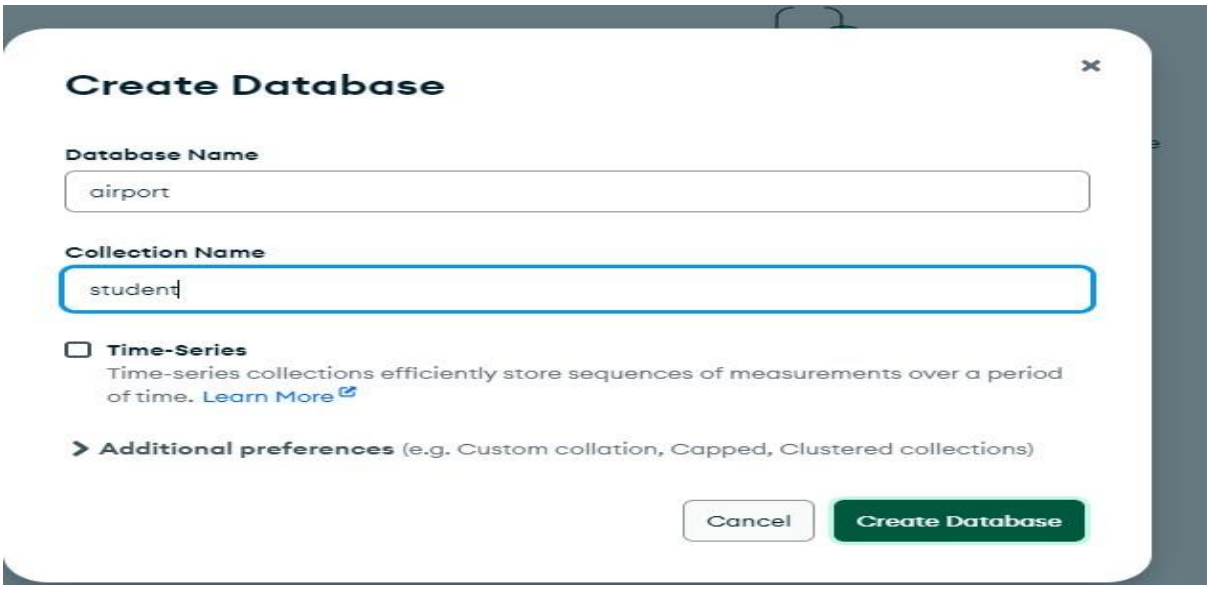
Step 3: Create a databases and upload airport file



No saved queries yet.

Start saving your aggregations and find queries, you'll see them here.

Not sure where to start? Visit our Docs →



Create Database

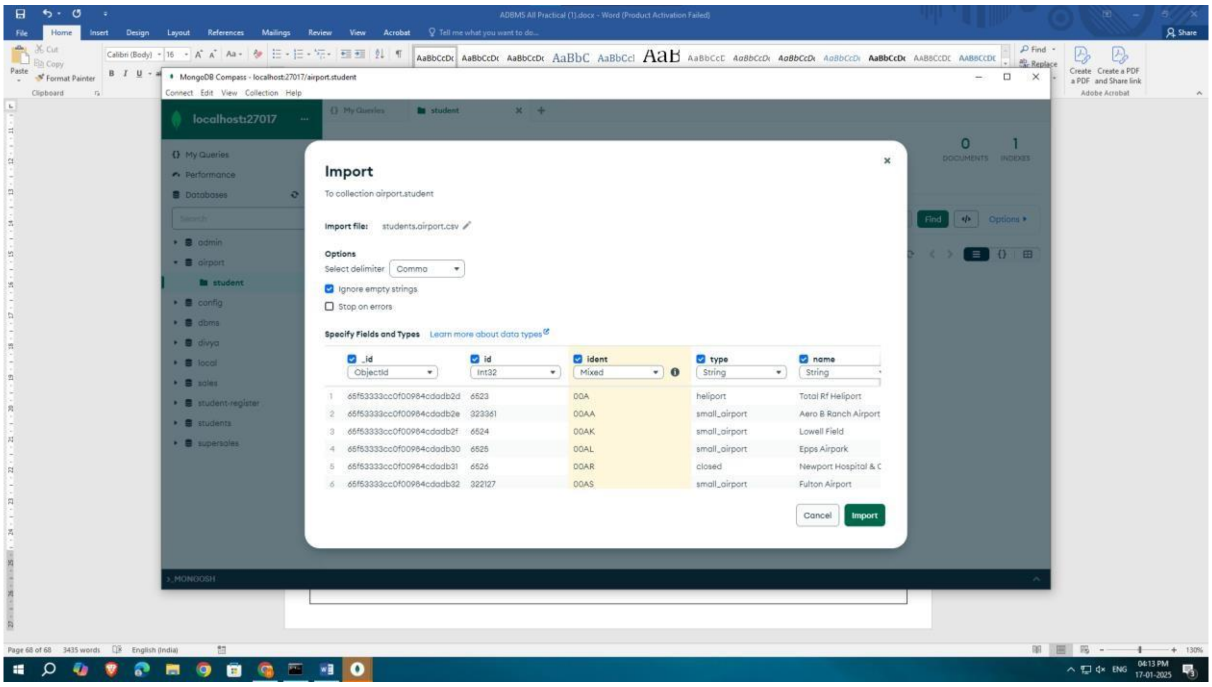
Database Name: airport

Collection Name: student

☐ 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)

Step 4: Import data and click on import


Import

To collection airport.student

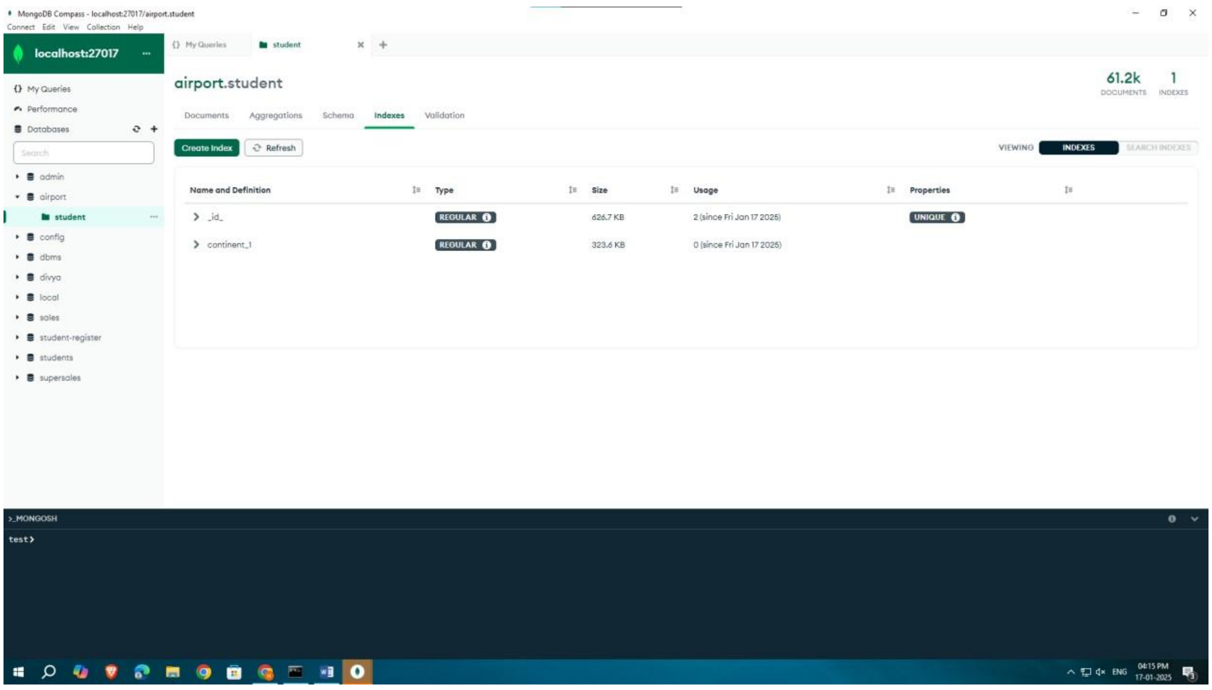
Import file: students.airport.csv

Options
Select delimiter: Comma
☑ Ignore empty strings
☐ Stop on errors

Specify Fields and Types

	_id (ObjectId)	id (Int32)	ident (Mixed)	type (String)	name (String)
1	65f53333cc0f00984cdadb2d	6523	00A	heliport	Total Rf Heliport
2	65f53333cc0f00984cdadb2e	323361	00AA	small_airport	Aero B Ranch Airport
3	65f53333cc0f00984cdadb2f	6524	00AK	small_airport	Lowell Field
4	65f53333cc0f00984cdadb30	6525	00AL	small_airport	Epps Airpark
5	65f53333cc0f00984cdadb31	6526	00AR	closed	Newport Hospital & C
6	65f53333cc0f00984cdadb32	322127	00AS	small_airport	Fulton Airport

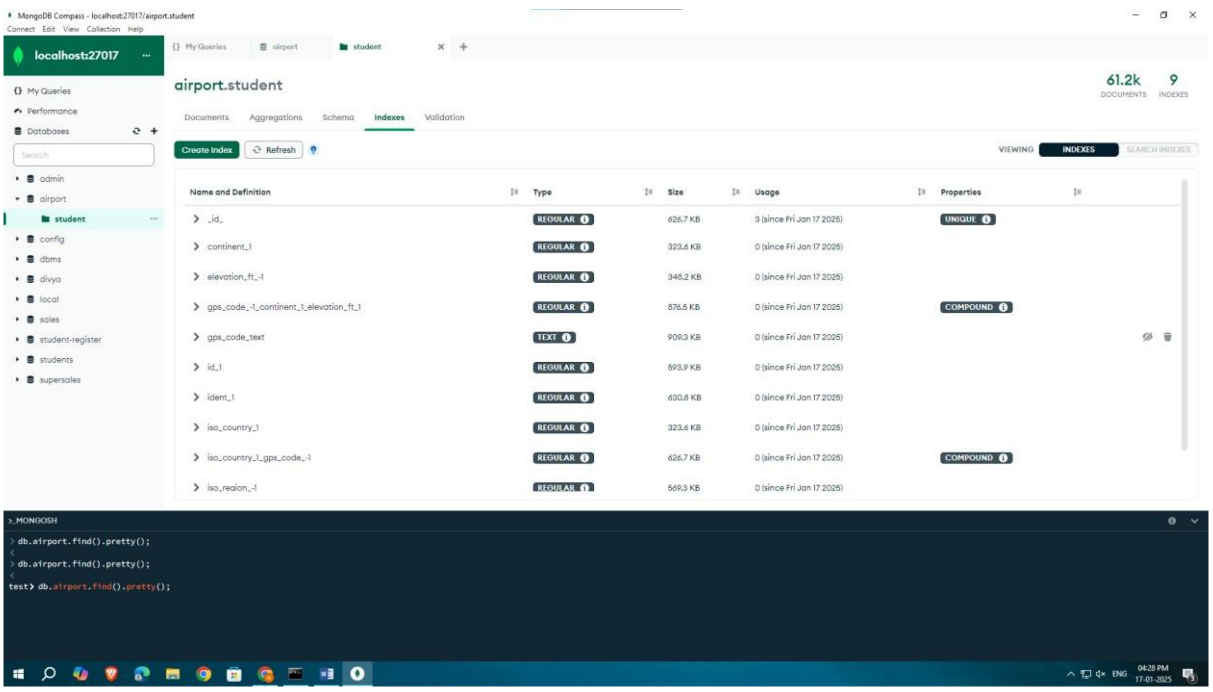
Step 5: Create Indexes


airport.student — 61.2k DOCUMENTS, 1 INDEXES

Name and Definition	Type	Size	Usage	Properties
_id_	REGULAR	626.7 KB	2 (since Fri Jan 17 2025)	UNIQUE
continent_1	REGULAR	323.6 KB	0 (since Fri Jan 17 2025)	

```
test>
```

Step 6: Using different indexes



airport.student

Name and Definition	Type	Size	Usage	Properties
_id_	REGULAR	626.7 KB	3 (since Fri Jan 17 2025)	UNIQUE
continent_1	REGULAR	323.6 KB	0 (since Fri Jan 17 2025)	
elevation_ft_-1	REGULAR	348.2 KB	0 (since Fri Jan 17 2025)	
gps_code_-1_continent_1_elevation_ft_1	REGULAR	876.5 KB	0 (since Fri Jan 17 2025)	COMPOUND
gps_code_text	TEXT	909.3 KB	0 (since Fri Jan 17 2025)	
id_1	REGULAR	593.9 KB	0 (since Fri Jan 17 2025)	
ident_1	REGULAR	630.8 KB	0 (since Fri Jan 17 2025)	
iso_country_1	REGULAR	323.6 KB	0 (since Fri Jan 17 2025)	
iso_country_1_gps_code_-1	REGULAR	626.7 KB	0 (since Fri Jan 17 2025)	COMPOUND
iso_region_-1	REGULAR	569.3 KB	0 (since Fri Jan 17 2025)	

```
>_MONGOSH
> db.airport.find().pretty();
<
> db.airport.find().pretty();
<
test> db.airport.find().pretty();
```