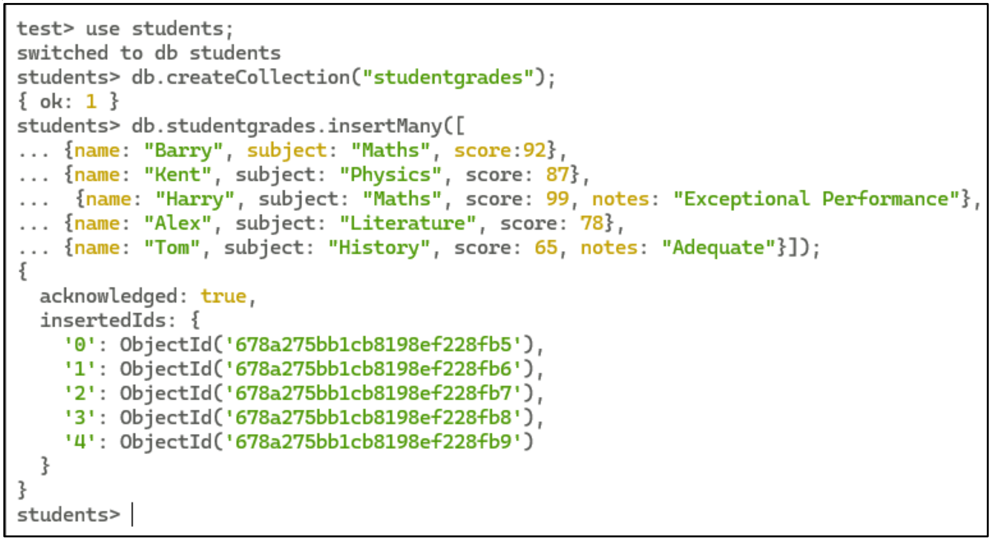
Mithibai College
Department of Computer Science

M.Sc. (Data Science and AI)
Practical 4: INDEXING USING MongoDB

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

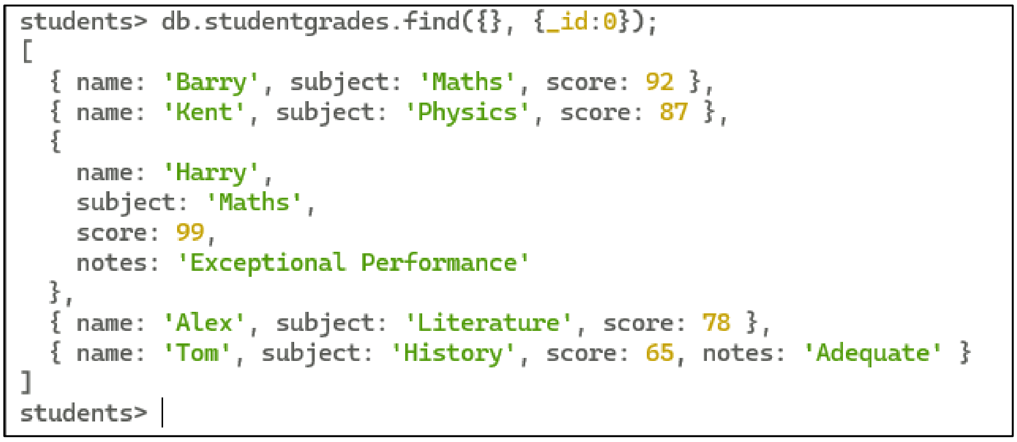
1. Mongo DB indexing
a. Create index in Mongo DB
b. Finding the indexes in a collection
c. Drop indexes in a collection
d. Drop all the indexes

```
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"}
 ]
)
```



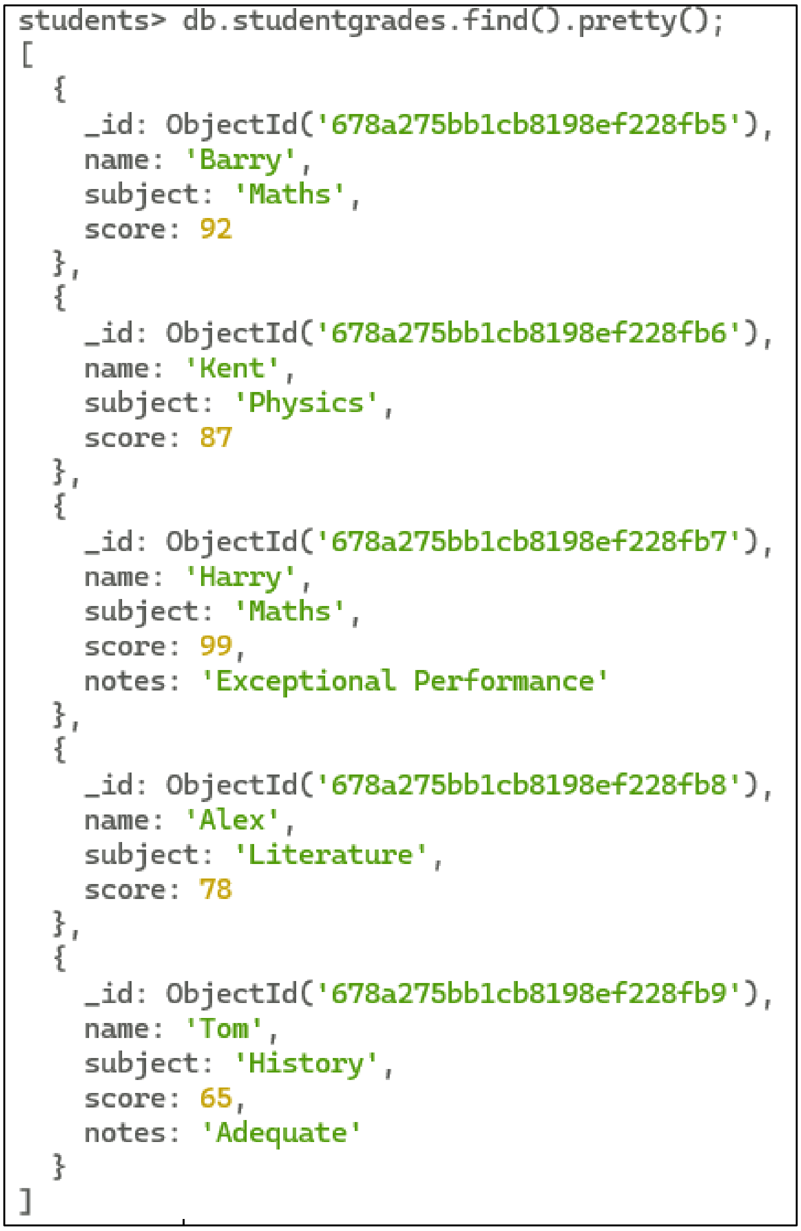
```
test> use students;
switched to db students
students> db.createCollection("studentgrades");
{ ok: 1 }
students> db.studentgrades.insertMany([
... {name: "Barry", subject: "Maths", score:92},
... {name: "Kent", subject: "Physics", score: 87},
...  {name: "Harry", subject: "Maths", score: 99, notes: "Exceptional Performance"},
... {name: "Alex", subject: "Literature", score: 78},
... {name: "Tom", subject: "History", score: 65, notes: "Adequate"}]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('678a275bb1cb8198ef228fb5'),
    '1': ObjectId('678a275bb1cb8198ef228fb6'),
    '2': ObjectId('678a275bb1cb8198ef228fb7'),
    '3': ObjectId('678a275bb1cb8198ef228fb8'),
    '4': ObjectId('678a275bb1cb8198ef228fb9')
  }
}
students> |
```

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



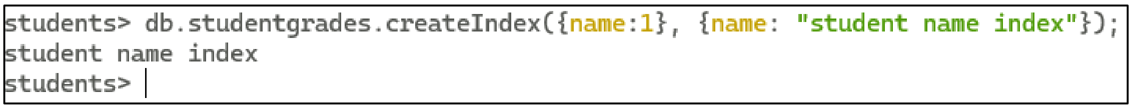
```
students> db.studentgrades.find({}, {_id:0});
[
  { name: 'Barry', subject: 'Maths', score: 92 },
  { name: 'Kent', subject: 'Physics', score: 87 },
  {
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  { name: 'Alex', subject: 'Literature', score: 78 },
  { name: 'Tom', subject: 'History', score: 65, notes: 'Adequate' }
]
students> |
```

db.studentgrades.find().pretty()



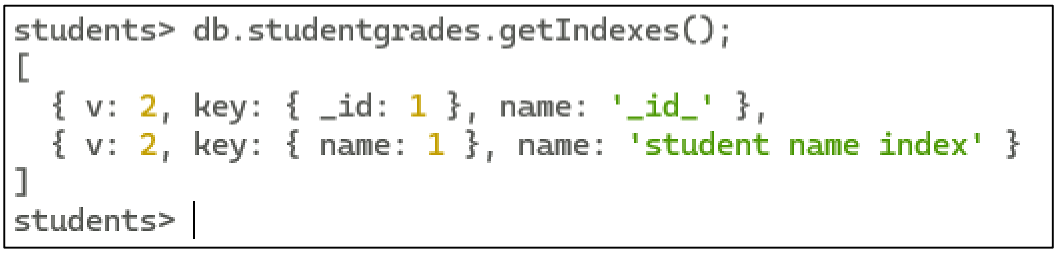
```
students> db.studentgrades.find().pretty();
[
  {
    _id: ObjectId('678a275bb1cb8198ef228fb5'),
    name: 'Barry',
    subject: 'Maths',
    score: 92
  },
  {
    _id: ObjectId('678a275bb1cb8198ef228fb6'),
    name: 'Kent',
    subject: 'Physics',
    score: 87
  },
  {
    _id: ObjectId('678a275bb1cb8198ef228fb7'),
    name: 'Harry',
    subject: 'Maths',
    score: 99,
    notes: 'Exceptional Performance'
  },
  {
    _id: ObjectId('678a275bb1cb8198ef228fb8'),
    name: 'Alex',
    subject: 'Literature',
    score: 78
  },
  {
    _id: ObjectId('678a275bb1cb8198ef228fb9'),
    name: 'Tom',
    subject: 'History',
    score: 65,
    notes: 'Adequate'
  }
]
```

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



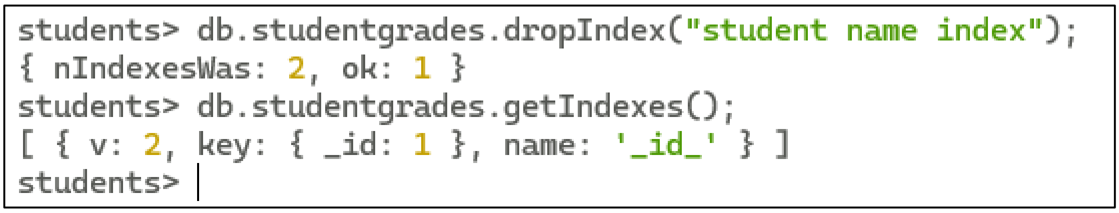
```
students> db.studentgrades.createIndex({name:1}, {name: "student name index"});
student name index
students> 
```

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()



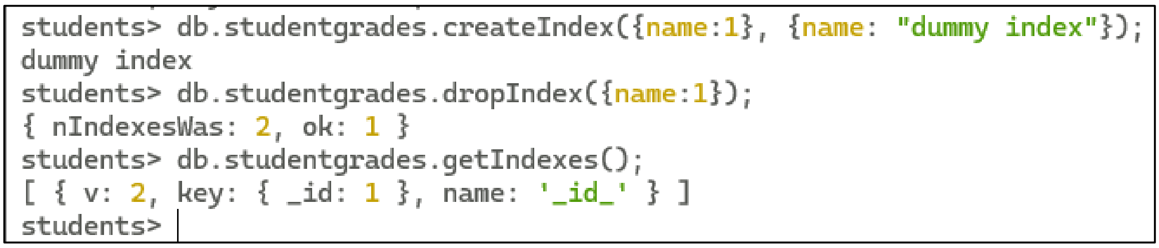
```
students> db.studentgrades.getIndexes();
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'student name index' }
]
students> 
```

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")



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

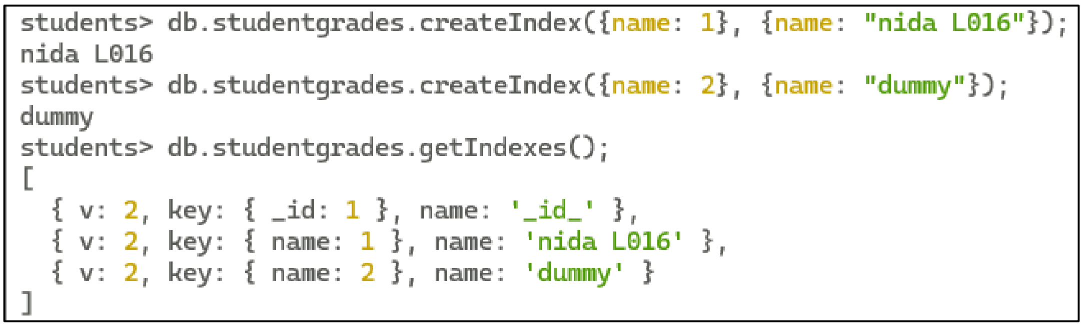
You can also use the index field value for removing an index without a defined name:
db.studentgrades.dropIndex({name:1})



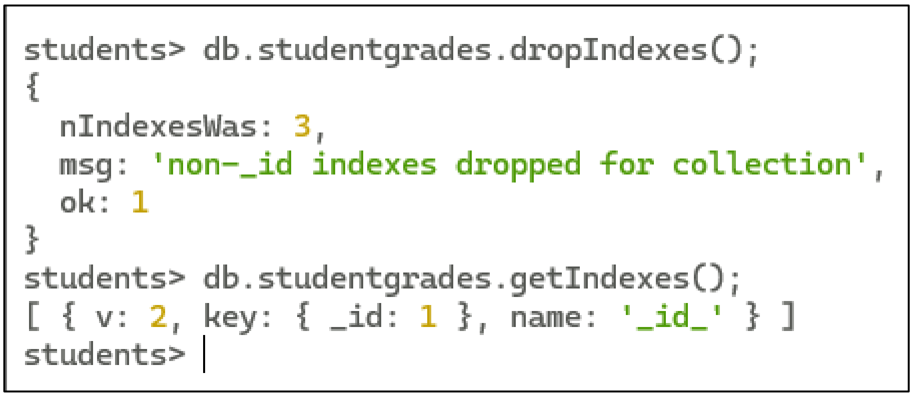
```
students> db.studentgrades.createIndex({name:1}, {name: "dummy index"});
dummy index
students> db.studentgrades.dropIndex({name:1});
{ nIndexesWas: 2, ok: 1 }
students> db.studentgrades.getIndexes();
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
students> 
```

The dropIndexes command can also drop all the indexes excluding the default _id index.

db.studentgrades.dropIndexes()

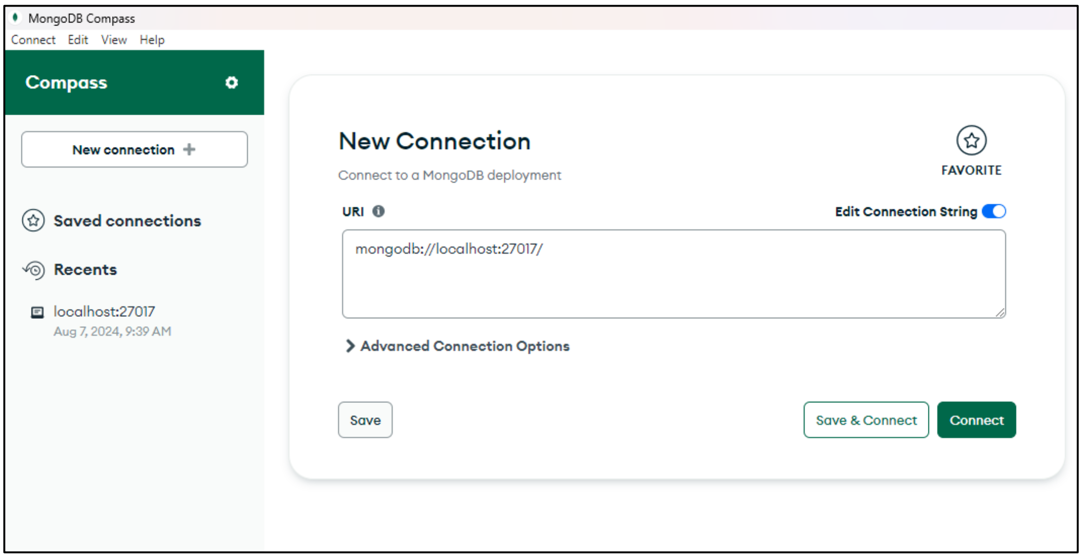


```
students> db.studentgrades.createIndex({name: 1}, {name: "nida L016"});
nida L016
students> db.studentgrades.createIndex({name: 2}, {name: "dummy"});
dummy
students> db.studentgrades.getIndexes();
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { name: 1 }, name: 'nida L016' },
  { v: 2, key: { name: 2 }, name: 'dummy' }
]
```



```
students> db.studentgrades.dropIndexes();
{
  nIndexesWas: 3,
  msg: 'non-_id indexes dropped for collection',
  ok: 1
}
students> db.studentgrades.getIndexes();
[ { v: 2, key: { _id: 1 }, name: '_id_' } ]
students> 
```

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



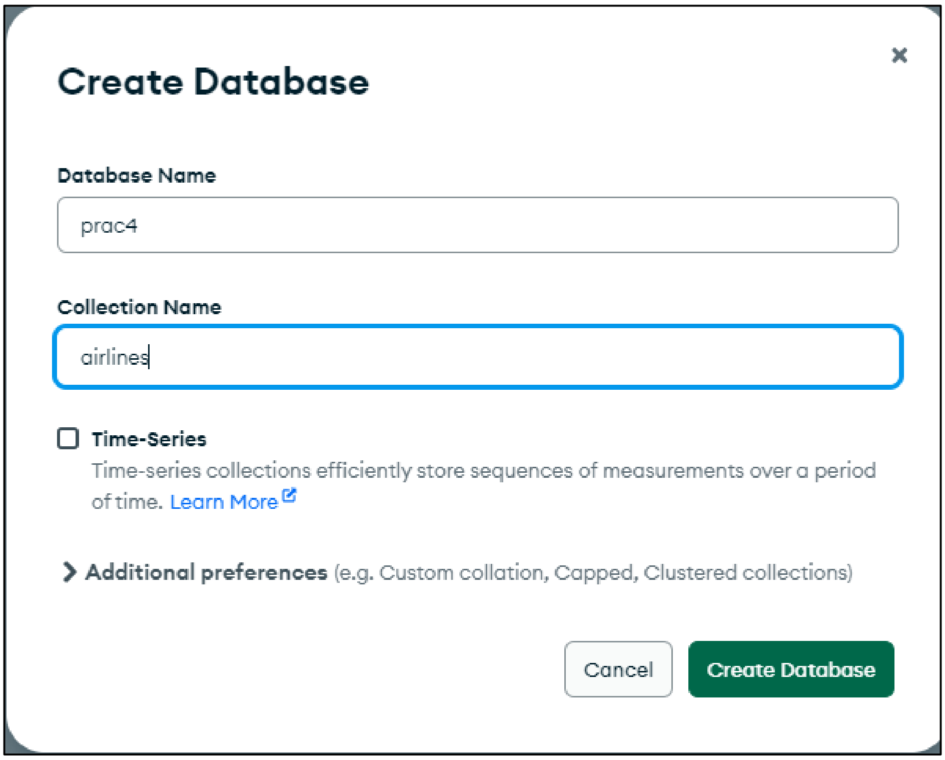
New Connection

Connect to a MongoDB deployment

URI

mongodb://localhost:27017/

> Advanced Connection Options



Create Database

Database Name

prac4

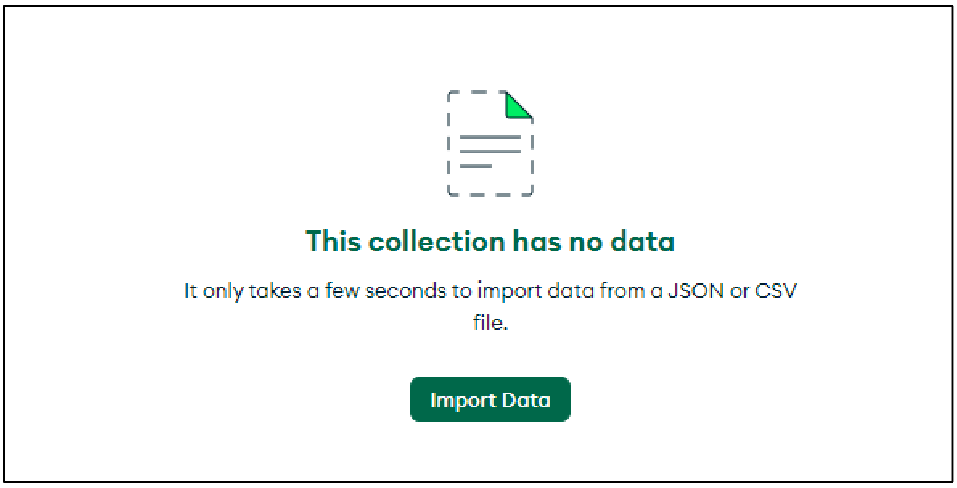
Collection Name

airlines

□ Time-Series

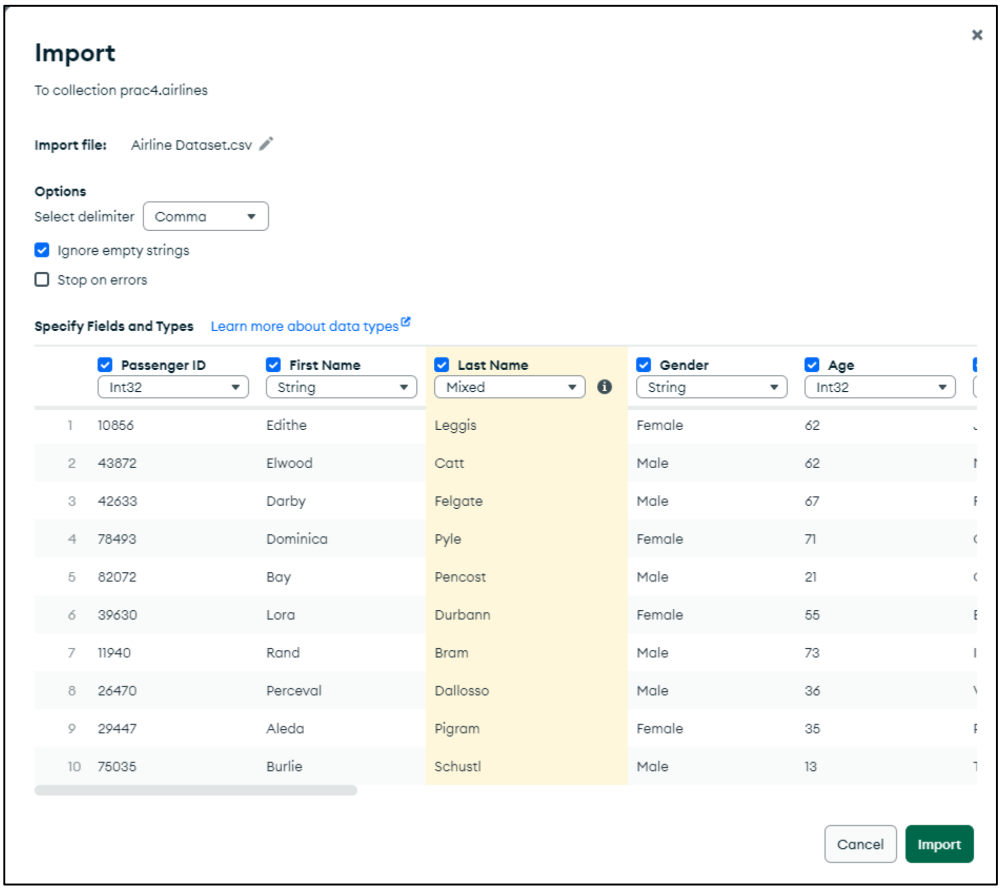
Time-series collections efficiently store sequences of measurements over a period of time. Learn More

> Additional preferences (e.g. Custom collation, Capped, Clustered collections)



This collection has no data

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



Import

To collection prac4.airlines

Import file: Airline Dataset.csv

Options

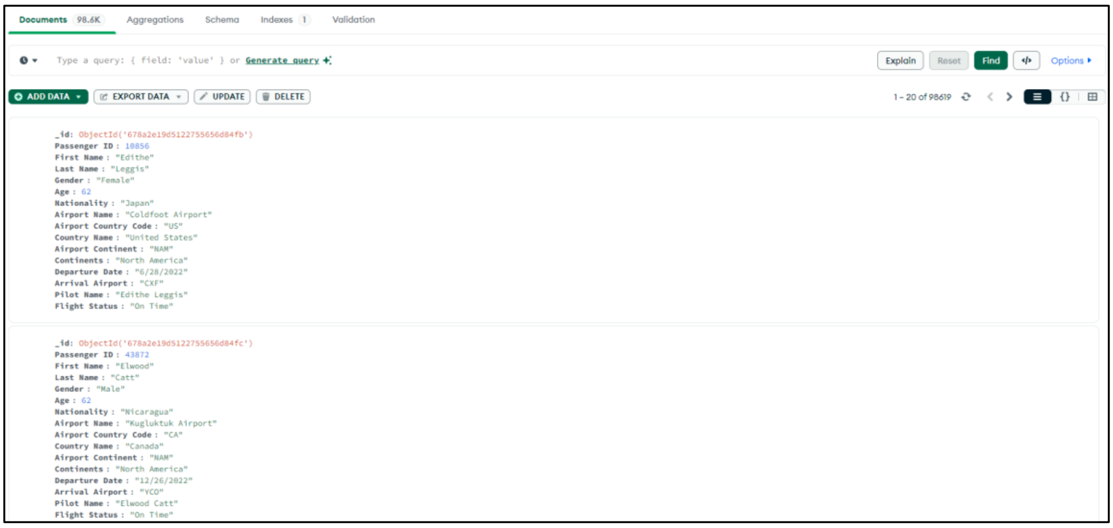
Select delimiter: Comma

☑ Ignore empty strings

□ Stop on errors

Specify Fields and Types Learn more about data types

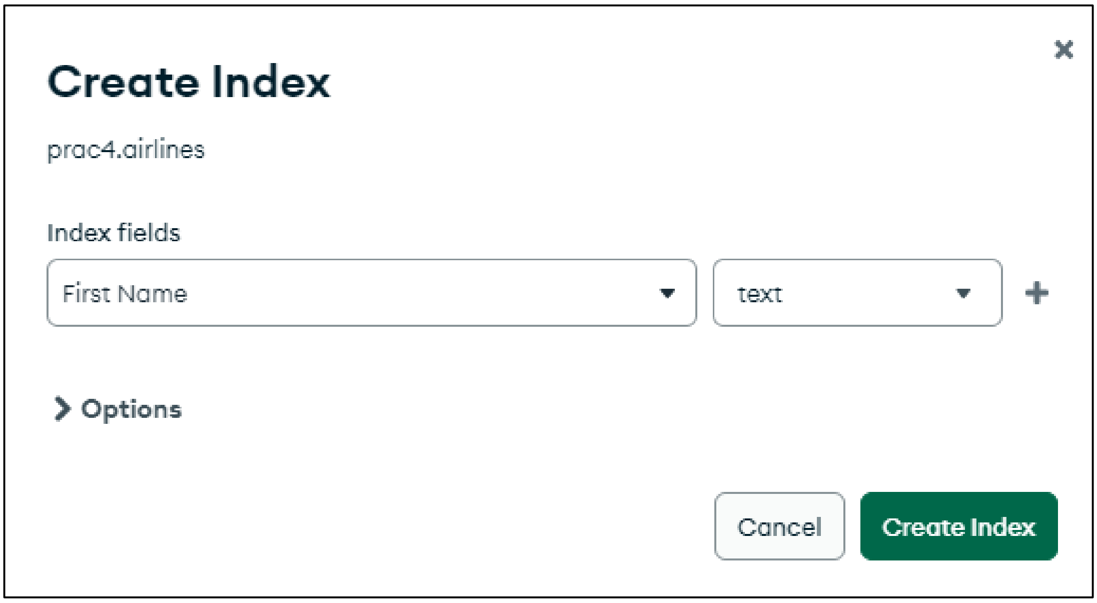
	☑ Passenger ID Int32	☑ First Name String	☑ Last Name Mixed	☑ Gender String	☑ Age Int32
1	10856	Edithe	Leggis	Female	62
2	43872	Elwood	Catt	Male	62
3	42633	Darby	Felgate	Male	67
4	78493	Dominica	Pyle	Female	71
5	82072	Bay	Pencost	Male	21
6	39630	Lora	Durbann	Female	55
7	11940	Rand	Bram	Male	73
8	26470	Perceval	Dallosso	Male	36
9	29447	Aleda	Pigram	Female	35
10	75035	Burlie	Schustl	Male	13



```
_id: ObjectId('678a2e19d5122755656d84fb')
Passenger ID : 10856
First Name : "Edithe"
Last Name : "Leggis"
Gender : "Female"
Age : 62
Nationality : "Japan"
Airport Name : "Coldfoot Airport"
Airport Country Code : "US"
Country Name : "United States"
Airport Continent : "NAM"
Continents : "North America"
Departure Date : "6/28/2022"
Arrival Airport : "CXF"
Pilot Name : "Edithe Leggis"
Flight Status : "On Time"

_id: ObjectId('678a2e19d5122755656d84fc')
Passenger ID : 43872
First Name : "Elwood"
Last Name : "Catt"
Gender : "Male"
Age : 62
Nationality : "Nicaragua"
Airport Name : "Kugluktuk Airport"
Airport Country Code : "CA"
Country Name : "Canada"
Airport Continent : "NAM"
Continents : "North America"
Departure Date : "12/26/2022"
Arrival Airport : "YCO"
Pilot Name : "Elwood Catt"
Flight Status : "On Time"
```

Single Valued Index

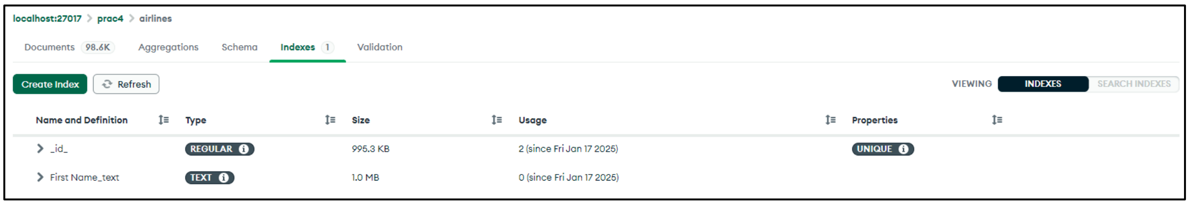


Create Index

prac4.airlines

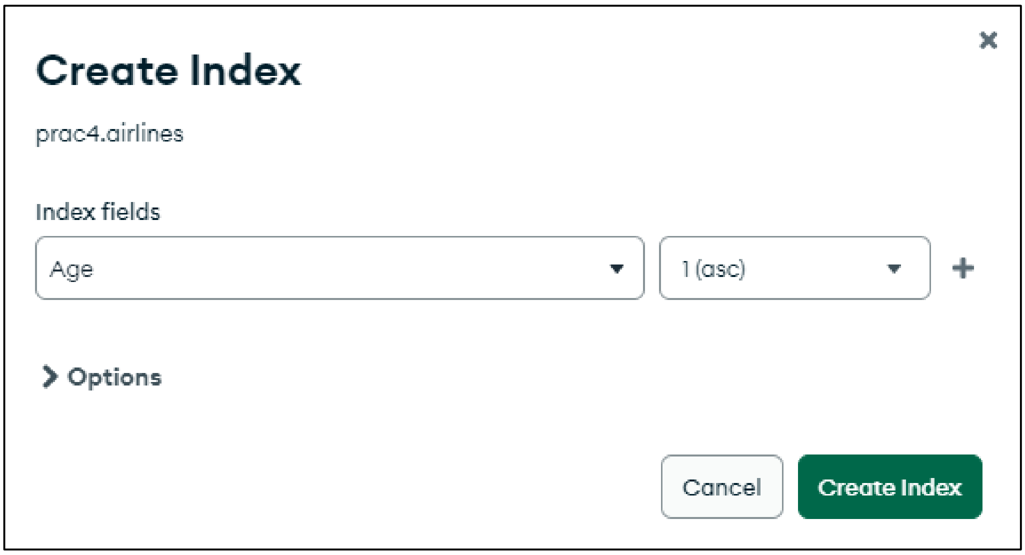
Index fields

First Name ▾ text ▾



localhost:27017 > prac4 > airlines

Name and Definition	Type	Size	Usage	Properties
> _id_	REGULAR	995.3 KB	2 (since Fri Jan 17 2025)	UNIQUE
> First Name_text	TEXT	1.0 MB	0 (since Fri Jan 17 2025)	

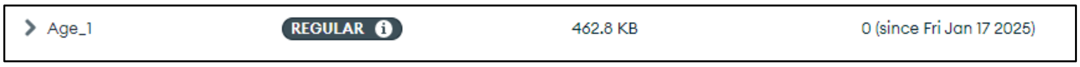


Create Index

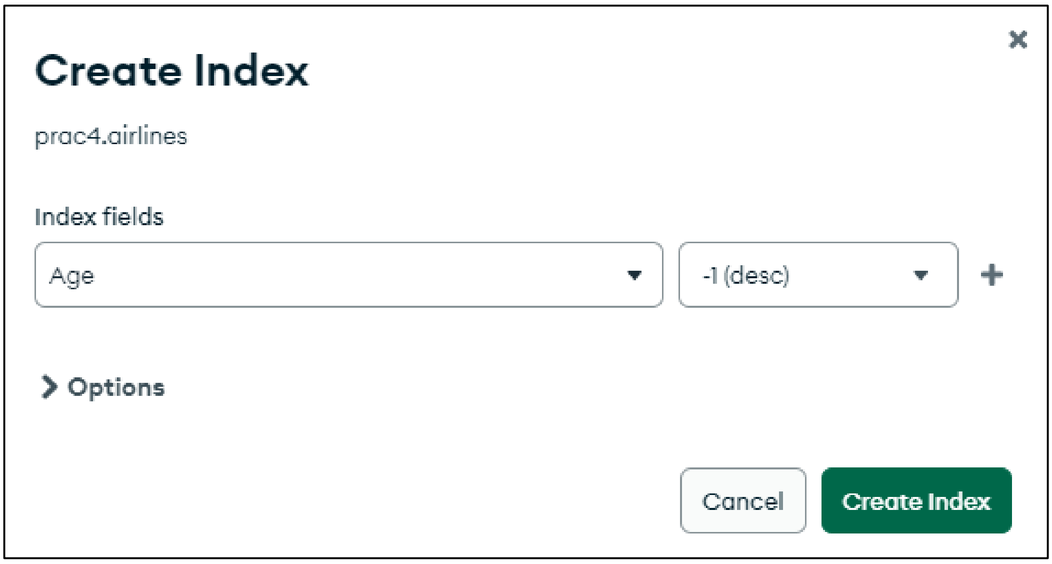
prac4.airlines

Index fields

Age 1 (asc)



Age_1 REGULAR 462.8 KB 0 (since Fri Jan 17 2025)

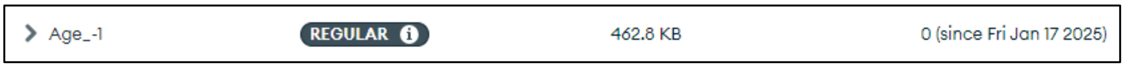


Create Index

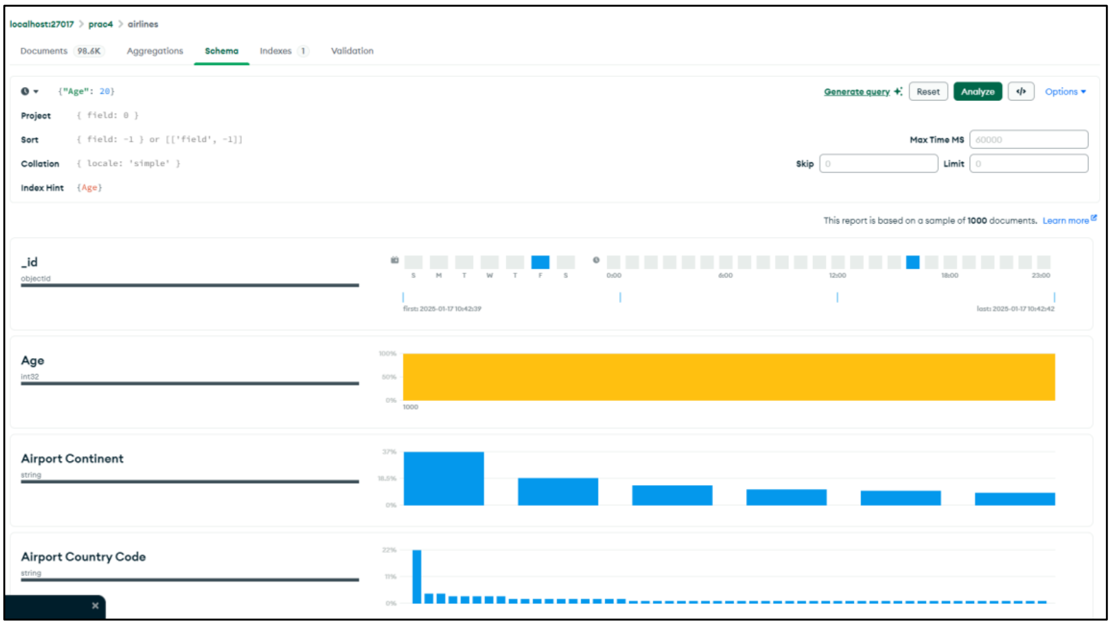
prac4.airlines

Index fields

Age -1 (desc)



Age_-1 REGULAR 462.8 KB 0 (since Fri Jan 17 2025)



localhost:27017 > prac4 > airlines

Documents 98.6K | Aggregations | Schema | Indexes 1 | Validation

{"Age": 20}

Project { field: 0 }

Sort { field: -1 } or [['field', -1]]

Collation { locale: 'simple' }

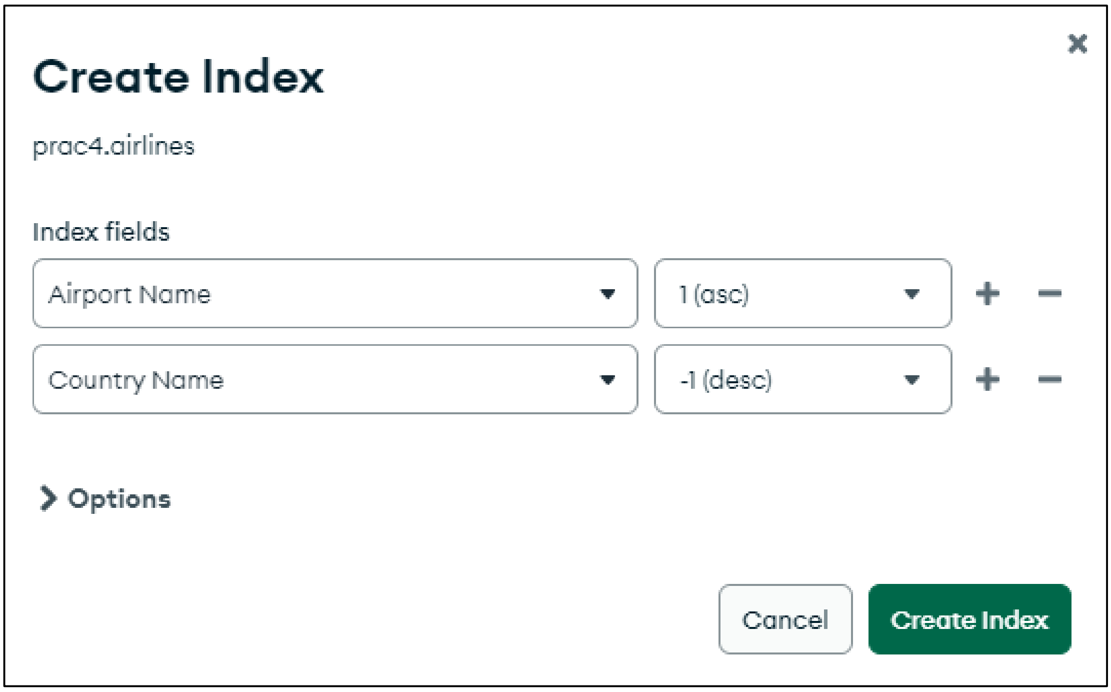
Index Hint {Age}

Max Time MS 60000

Skip 0 Limit 0

This report is based on a sample of **1000** documents. Learn more

Compound Index

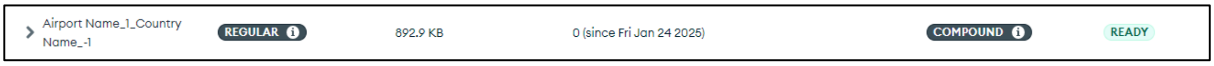


Create Index

prac4.airlines

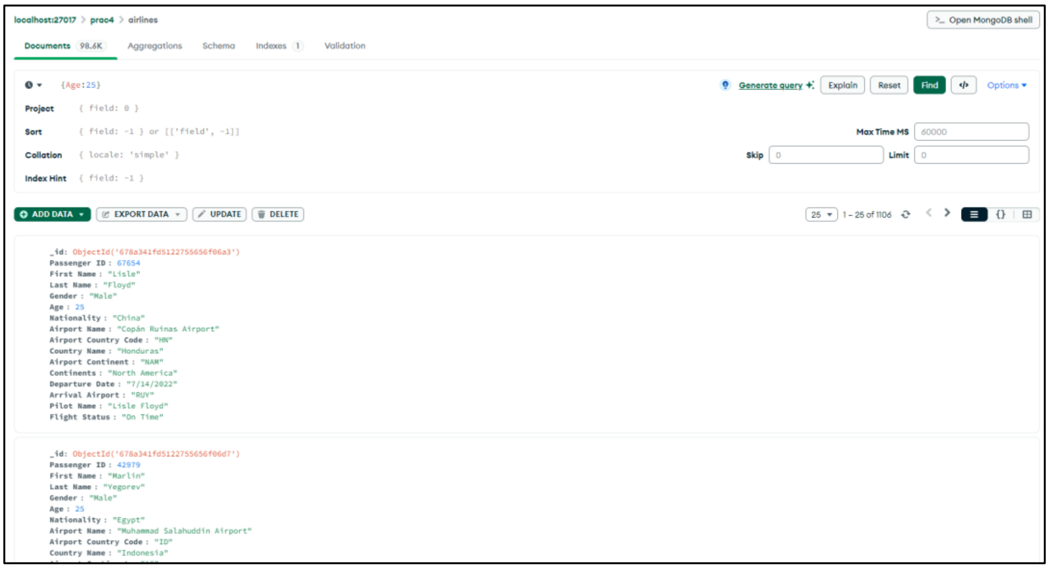
Index fields

Airport Name	1 (asc)
Country Name	-1 (desc)



Airport Name_1_Country Name_-1 REGULAR 892.9 KB 0 (since Fri Jan 24 2025) COMPOUND READY

Difference in query performance after creating index



{Age:25}

Project: { field: 0 }

Sort: { field: -1 } or [['field', -1]]

Collation: { locale: 'simple' }

Index Hint: { field: -1 }

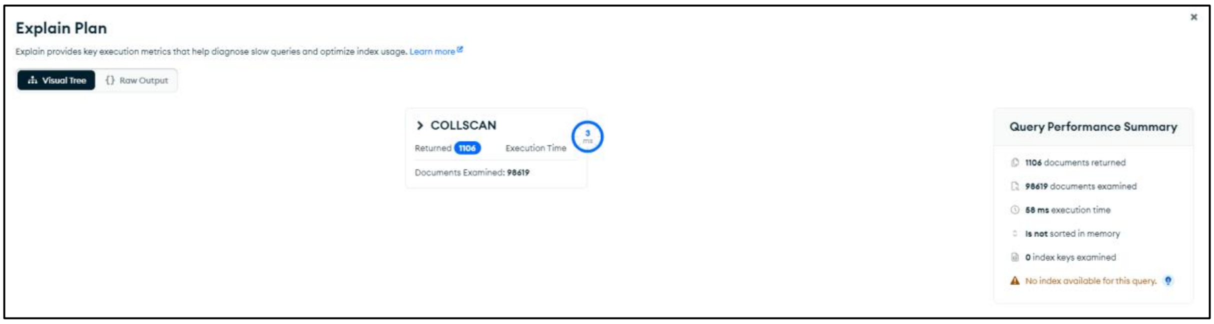
Max Time MS: 60000

Skip: 0

Limit: 0

```
_id: ObjectId('678a341fd5122755656f06a3')
Passenger ID : 67654
First Name : "Lisle"
Last Name : "Floyd"
Gender : "Male"
Age : 25
Nationality : "China"
Airport Name : "Copán Ruinas Airport"
Airport Country Code : "HN"
Country Name : "Honduras"
Airport Continent : "NAM"
Continents : "North America"
Departure Date : "7/14/2022"
Arrival Airport : "RUY"
Pilot Name : "Lisle Floyd"
Flight Status : "On Time"
```

```
_id: ObjectId('678a341fd5122755656f06d7')
Passenger ID : 42979
First Name : "Marlin"
Last Name : "Yegorev"
Gender : "Male"
Age : 25
Nationality : "Egypt"
Airport Name : "Muhammad Salahuddin Airport"
Airport Country Code : "ID"
Country Name : "Indonesia"
```



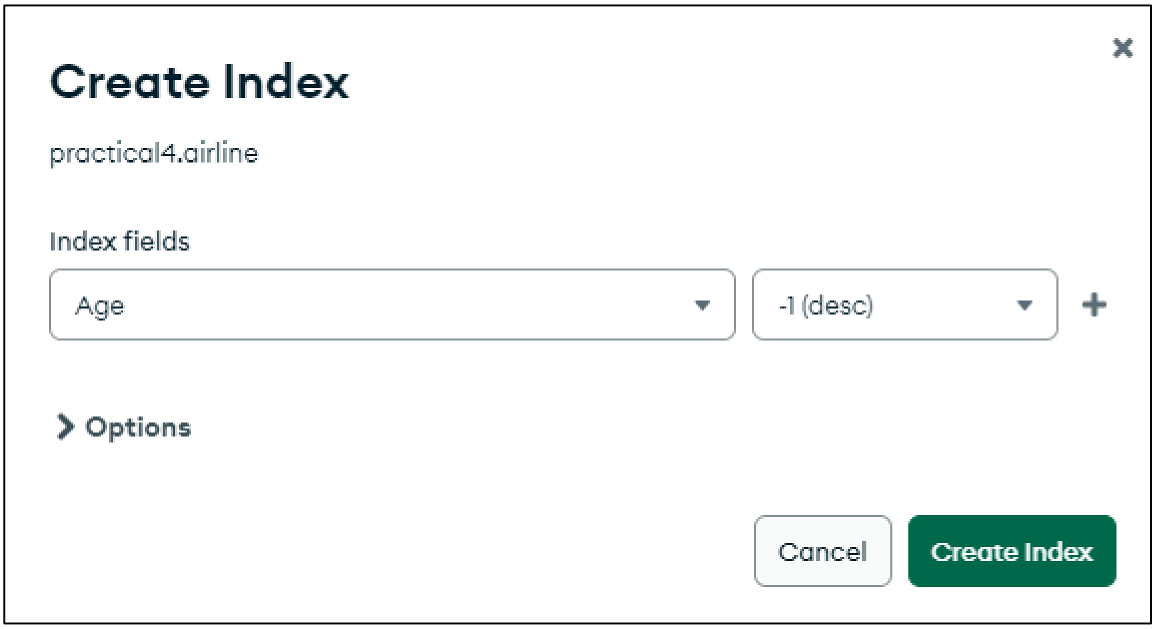
Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. Learn more

Documents Examined: 98619

Query Performance Summary

- 1106 documents returned
- 98619 documents examined
- 58 ms execution time
- Is not sorted in memory
- 0 index keys examined
- No index available for this query.

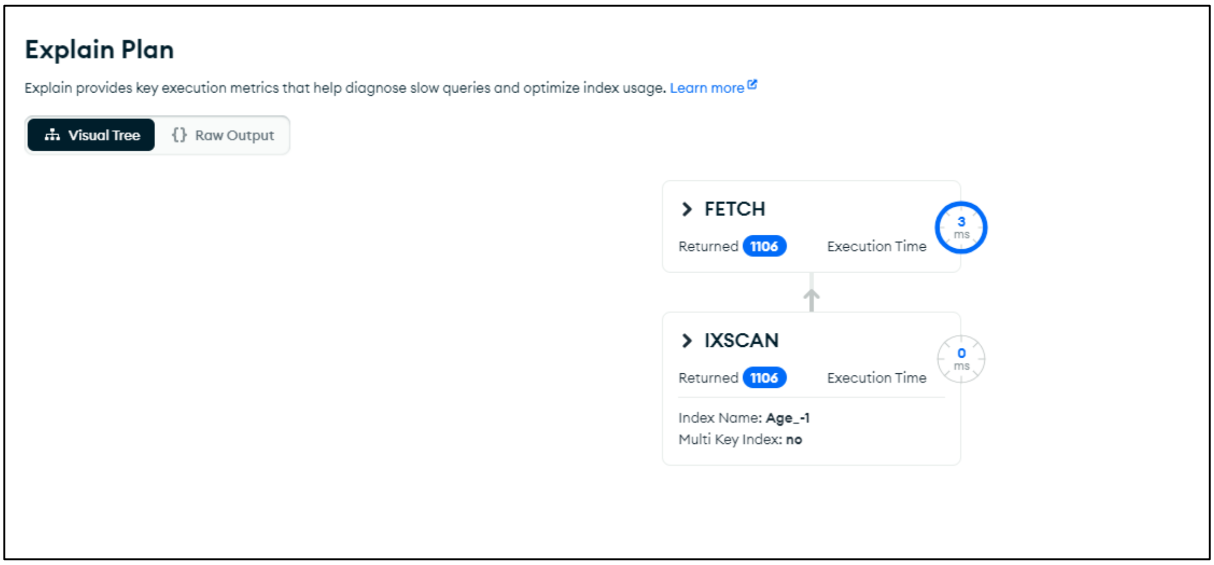


Create Index

practical4.airline

Index fields

Age -1 (desc)



Explain Plan

Explain provides key execution metrics that help diagnose slow queries and optimize index usage. Learn more

FETCH

Returned 1106 Execution Time 3 ms

IXSCAN

Returned 1106 Execution Time 0 ms

Index Name: Age_-1

Multi Key Index: no