

## Practical - 4

### INDEXING USING MongoDB

17-01-25

Write-up: -

- Indexing in mongodb
- Types of indexing
- Document document-oriented NoSQL

#### Q1. MongoDB indexing

##### a. Create index in MongoDB

```
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('678a268b9966c85063b51856'),  
  '1': ObjectId('678a268b9966c85063b51857'),  
  '2': ObjectId('678a268b9966c85063b51858'),  
  '3': ObjectId('678a268b9966c85063b51859'),  
  '4': ObjectId('678a268b9966c85063b5185a')  
}  
}
```

#### b. Finding the indexes in a collection

```
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('678a268b9966c85063b51856'),
name: 'Barry',
subject: 'Maths',
score: 92
},
{
  _id: ObjectId('678a268b9966c85063b51857'),
  name: 'Kent',
  subject: 'Physics',
  score: 87
},
{
  _id: ObjectId('678a268b9966c85063b51858'),
  name: 'Harry',
  subject: 'Maths',
  score: 99,
  notes: 'Exceptional Performance'
},
{
  _id: ObjectId('678a268b9966c85063b51859'),
  name: 'Alex',
  subject: 'Literature',
  score: 78
},
{
  _id: ObjectId('678a268b9966c85063b5185a'),
  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_' },
  { v: 2, key: { name: 1 }, name: 'student name index' }
]
```

#### c. Drop indexes in a collection

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

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

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

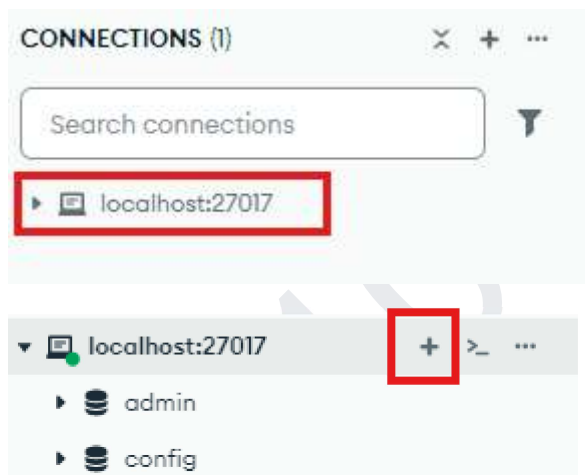
#### d. Drop all the indexes

```
students> db.studentgrades.dropIndexes()
```

```
{  
  nIndexesWas: 1,  
  msg: 'non-_id indexes dropped for collection',  
  ok: 1  
}
```

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

Q2) Create all the types of indexes (discussed in class) which will help in finding certain words in a document by using AIRPORT (dataset).



## Create Database

Database Name

practical\_3

Collection Name

airport

☐ 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

flights.csv

raw-flight-data.csv

airports.csv

## Import

To collection practical\_3.airport

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 Int32	<input checked="" type="checkbox"/> city String	<input checked="" type="checkbox"/> state String	<input checked="" type="checkbox"/> name 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

Documents 365 Aggregations Schema **Indexes 1** Validation

Create Index

Refresh

VIEWING

INDEXES

SEARCH INDEXES

Name & Definition	Type	Size	Usage	Properties	Status
> _id_	REGULAR	20.5 KB	2 (since Fri Jan 17 2025)	UNIQUE	READY

Documents 365

Aggregations

Schema

Indexes 1

Validation

Create Index

Refresh

Name & Definition

Type

Size

Usage

> \_id\_

REGULAR

20.5 KB

3 (since Fri Jan 17 2025)

## Create Index

practical\_3.airport

Index fields

airport\_id

1 (asc)

+

> Options

Cancel

Create Index

## Create Index

practical\_3.airport

Index fields

city

-1 (desc)

+

> Options

Cancel

Create Index



## Create Index

practical\_3.airport

Index fields

name

text

+

> Options

Cancel

Create Index

Name & Definition	Type	Size	Usage	Properties	Status
> _id_	REGULAR	20.5 KB	3 (since Fri Jan 17 2025)	UNIQUE	READY
> airport_id_1	REGULAR	20.5 KB	0 (since Fri Jan 17 2025)		READY
> city_-1	REGULAR	24.6 KB	0 (since Fri Jan 17 2025)		READY
> name_text	TEXT	41.0 KB	0 (since Fri Jan 17 2025)		READY

localhost:27017 > practical\_3 > airport

Documents 365

Aggregations

Schema

Indexes 1

Validation

Type a query: { field: 'value' } or [Generate query](#)

+ ADD DATA

EXPORT DATA

UPDATE

DELETE

Documents 365 Aggregations Schema Indexes 4 Validation

`{ state: "AK" }` [Generate query](#) [Reset](#) [Analyze](#)

**Project** { field: 0 }

**Sort** { field: -1 } or [['field', -1]] **Max Time MS** 60000

**Collation** { locale: 'simple' } **Skip** 0 **Limit** 0

**Index Hint** airport\_id

---

**\_id**  
objectId

Inserted: 2025-01-17 09:59:20

---

**airport\_id**  
int32

13873 15841 14828 12819 10754 10245 14256 11445 11336 10170 15991 10304 10926 10561 14709 10165 12623 11997 10299 12807

---

**city**  
string

Deadhorse Anchorage Cordova Petersburg Aniak Juneau Nome Ketchikan Wrangell St. Mary's Unalaska King Salmon Fairbanks Bethel Yakutat Dillingham Sitka Barrow Adak Island Kodiak

---

**name**  
string

Aniak Airport Ketchikan International Deadhorse Airport Petersburg James A Johnson Dillingham Airport Adak Bethel Airport Juneau International Nome Airport King Salmon Airport Sitka Rocky Gutierrez Wiley Post/Will Rogers Memorial Kodiak Airport St. Mary's Airport Ralph Wien Memorial Wrangell Airport Gustavus Airport Unalaska Airport Yakutat Airport Fairbanks International