Nikit Gokhe

Class- Comp D1

Roll No. 224024

GR No. 21810522

**ASSIGNMENT NO.08**

**AIM :** Implement aggregation and indexing with suitable example using MongoDB.

**OBJECTIVE :** To implement aggregation and indexing with suitable example using MongoDB.

**THEORY** **:**

**Aggregation:**

Aggregation in MongoDB is nothing but an operation used to process the data that returns the computed results. Aggregation basically groups the data from multiple documents and operates in many ways on those grouped data in order to return one combined result. In sql count(\*) and with group by is an equivalent of MongoDB aggregation.

Aggregate function groups the records in a collection, and can be used to provide total number(sum), average, minimum, maximum etc out of the group selected.

In order to perform the aggregate function in MongoDB, aggregate () is the function to be used.

### ***Syntax:***

db.collection\_name.aggregate(aggregate\_operation)

## Different expressions used by Aggregate function

|  |  |
| --- | --- |
| Expression | Description |
| $sum | Summates the defined values from all the documents in a collection |
| $avg | Calculates the average values from all the documents in a collection |
| $min | Return the minimum of all values of documents in a collection |
| $max | Return the maximum of all values of documents in a collection |
| $addToSet | Inserts values to an array but no duplicates in the resulting document |
| $push | Inserts values to an array in the resulting document |
| $first | Returns the first document from the source document |
| $last | Returns the last document from the source document |

**Indexes:**

Indexes in SQL programming are nothing but a special data structure used to easily and quickly locate the record in a given table of the database without being required to traverse through each and every record of the table. Indexes are easily generated using one or more columns of a given table. As a note, the data structure used by an index is a Binary Tree (B-Tree).

In MongoDB, indexes plays a vital role in efficiently execution of the queries. Basically, if no index is defined in MongoDB, then it has to scan every document of a given collection. Hence, MongoDB uses index to reduce the number of documents to be scanned in a given collection. In fact, MongoDB's index is more or less similar to the indexes used in other relational databases. The fact is that the MongoDB defines the indexes at the collection level and supports indexing on any fields in a MongoDB collection.

## Default Index in MongoDB

Mongodb provides a default index named \_id which acts as a primary key to access any document in a collection. This \_id index basically avoids the insertion of 2 documents with the same value for the \_id field.

### ***Syntax:***

db.collection\_name.createIndex({field : value })

### Types of Indexes in MongoDB

|  |  |
| --- | --- |
| **Index Type** | **Description** |
| Single field index | Used to create an index on a single field and it can be a user defined as well apart from the default \_id one. |
| Compound index | MongoDB supports the user-defined indexes on multiple fields. |
| Multi key index | MongoDB uses multi key indexes basically to store the arrays. MongoDB creates a separate index for each element in an array. MongoDB intelligently identifies to create a multi key index if the index contains elements from an array. |
| Geospatial index | Used to support the queries required for the geospatial coordinate data. |
| Text index | This index is used to search for a string content in a collection |
| Hashed index | Used for hash based Sharding |

**PROGRAM:**

**AGGREGATION**

> show dbs;

admin (empty)

computer 0.078GB

mymdb 0.078GB

t4 0.078GB

> use myddb

switched to db myddb

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CREATE COLLECTION:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CMD:db.createCollection("student");

{ "ok" : 1 }

CMD:

db.student.insert({Rollno:1,name:'Navin ',subject:'DMSA',marks:78});

WriteResult({ "nInserted" : 1 })

db.student.insert({Rollno:2,name:'anusha',subject:'OSD',marks:75});

WriteResult({ "nInserted" : 1 })

db.student.insert({Rollno:3,name:'ravi',subject:'TOC',marks:69});

WriteResult({ "nInserted" : 1 })

db.student.insert({Rollno:4,name:'veena',subject:'TOC',marks:70});

WriteResult({ "nInserted" : 1 })

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

DISPLAY RECORD:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CMD: db.student.find();

{ "\_id" : ObjectId("541bffb74ebca325353a6529"), "Rollno" : 1, "name" : "Navin", "subject" : "DMSA", "marks" : 78 }

{ "\_id" : ObjectId("541bffd34ebca325353a652a"), "Rollno" : 2, "name" : "anusha", "subject" : "OSD", "marks" : 75 }

{ "\_id" : ObjectId("541c00134ebca325353a652b"), "Rollno" : 3, "name" : "ravi", "subject" : "TOC", "marks" : 69 }

{ "\_id" : ObjectId("541c00134ebca325353a652b"), "Rollno" : 4, "name" : "veena", "subject" : "TOC", "marks" : 70 }

CMD:db.student.find().pretty();

{

"\_id" : ObjectId("541bffb74ebca325353a6529"),

"Rollno" : 1,

"name" : "Navin",

"subject" : "DMSA",

"marks" : 78

}

{

"\_id" : ObjectId("541bffd34ebca325353a652a"),

"Rollno" : 2,

"name" : "anusha",

"subject" : "OSD",

"marks" : 75

}

{

"\_id" : ObjectId("541c00134ebca325353a652b"),

"Rollno" : 3,

"name" : "ravi",

"subject" : "TOC",

"marks" : 69

}

{

"\_id" : ObjectId("541e00134ebca325353a652b"),

"Rollno" : 4,

"name" : "veena",

"subject" : "TOC",

"marks" : 70

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AGGREGATE FUNCTIONS:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MIN():

CMD: db.student.aggregate([{$group : {\_id : "$subject", marks : {$min : "$marks"}}}]);

{ "\_id" : "TOC", "marks" : 70 }

{ "\_id" : "OSD", "marks" : 75 }

{ "\_id" : "DMSA", "marks" : 78 }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MAX():

CMD:db.student.aggregate([{$group : {\_id : "$subject", marks : {$max : "$marks"}}}]);

{ "\_id" : "TOC", "marks" : 69 }

{ "\_id" : "OSD", "marks" : 75 }

{ "\_id" : "DMSA", "marks" : 78 }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

SUM():

CMD:db.student.aggregate([{$group : {\_id : "$subject", marks : {$sum : "$marks"}}}]);

{ "\_id" : "TOC", "marks" : 139 }

{ "\_id" : "OSD", "marks" : 75 }

{ "\_id" : "DMSA", "marks" : 78 }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AVG():

CMD:db.student.aggregate([{$group : {\_id : "$subject", marks : {$avg : "$marks"}}}]);

{ "\_id" : "TOC", "marks" : 69.5 }

{ "\_id" : "OSD", "marks" : 75 }

{ "\_id" : "DMSA", "marks" : 78 }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

FIRST():

CMD:db.student.aggregate([{$group : {\_id : "$subject", marks : {$first : "$marks"}}}]);

{ "\_id" : "TOC", "marks : 69 }

{ "\_id" : "OSD", "marks" : 75 }

{ "\_id" : "DMSA", "marks" : 78 }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

LAST():

CMD:db.student.aggregate([{$group : {\_id : "$subject", marks : {$last : "$marks"}}}]);

{ "\_id" : "TOC", "marks" : 70 }

{ "\_id" : "OSD", "marks" : 75 }

{ "\_id" : "DMSA", "marks" : 78 }

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INDEXES**

> show dbs

TE 0.078GB

admin (empty)

computer 0.078GB

gj 0.078GB

local 0.078GB

mydb 0.078GB

mymdb 0.078GB

t4 0.078GB

> use mydb

switched to db mydb

//CREATE COLLECTION:

> db.createCollection("student");

{ "ok" : 1 }

> db.student.insert({Rollno:1,name:'nikita',subject:'DMSA',marks:78});

WriteResult({ "nInserted" : 1 })

> db.student.insert({Rollno:2,name:'kiran',subject:'OSD',marks:75});

WriteResult({ "nInserted" : 1 })

> db.student.insert({Rollno:3,name:'komal',subject:'TOC',marks:69});

WriteResult({ "nInserted" : 1 })

> db.student.insert({Rollno:4,name:'snehal',subject:'TOC',marks:81});

WriteResult({ "nInserted" : 1 })

> db.student.insert({Rollno:5,name:'saurabh',subject:'TOC',marks:78});

WriteResult({ "nInserted" : 1 })

//DISPLAY COLLECTION:

> db.student.find();

{ "\_id" : ObjectId("541bffb74ebca325353a6529"), "Rollno" : 1, "name" : "nikita", "subject" : "DMSA", "marks" : 78 }

{ "\_id" : ObjectId("541bffd34ebca325353a652a"), "Rollno" : 2, "name" : "kiran", "subject" : "OSD", "marks" : 75 }

{ "\_id" : ObjectId("541c00134ebca325353a652b"), "Rollno" : 3, "name" : "komal", "subject" : "TOC", "marks" : 69 }

{ "\_id" : ObjectId("541c00274ebca325353a652c"), "Rollno" : 4, "name" : "snehal", "subject" : "TOC", "marks" : 81 }

{ "\_id" : ObjectId("541c007c4ebca325353a652d"), "Rollno" : 5, "name" : "saurabh", "subject" : "TOC", "marks" : 78 }

> db.student.find().pretty();

{

"\_id" : ObjectId("541bffb74ebca325353a6529"),

"Rollno" : 1,

"name" : "nikita",

"subject" : "DMSA",

"marks" : 78

}

{

"\_id" : ObjectId("541bffd34ebca325353a652a"),

"Rollno" : 2,

"name" : "kiran",

"subject" : "OSD",

"marks" : 75

}

{

"\_id" : ObjectId("541c00134ebca325353a652b"),

"Rollno" : 3,

"name" : "komal",

"subject" : "TOC",

"marks" : 69

}

{

"\_id" : ObjectId("541c0e05a8dc3dd4fa9f859e"),

"Rollno" : 4,

"name" : "snehal",

"subject" : "TOC",

"marks" : 81

}

{

"\_id" : ObjectId("541c0e2ca8dc3dd4fa9f85a1"),

"Rollno" : 5,

"name" : "saurabh",

"subject" : "TOC",

"marks" : 78

}

//USING ENSURE INDEX

> db.student.ensureIndex({Rollno:1});

{ "numIndexesBefore" : 3, "note" : "all indexes already exist", "ok" : 1 }

//USING GET INDEX:

> db.student.getIndexes();

[

{

"v" : 1,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "mydb.student"

},

{

"v" : 1,

"key" : {

"marks" : 1

},

"name" : "marks\_1",

"ns" : "mydb.student"

},

{

"v" : 1,

"key" : {

"Rollno" : 1

},

"name" : "Rollno\_1",

"ns" : "mydb.student"

}

// USING MINIMUM INDEX:

> db.student.find().min({Rollno:4});

{ "\_id" : ObjectId("541c0e05a8dc3dd4fa9f859e"), "Rollno" : 4, "name" : "snehal", "subject" : "TOC", "marks" : 81 }

{ "\_id" : ObjectId("541c0e05a8dc3dd4fa9f859f"), "Rollno" : 5, "name" : "saurabh", "subject" : "TOC", "marks" : 78 }

//USING MAXIMUM INDEX:

> db.student.find().max({Rollno:2});

{ "\_id" : ObjectId("541c0cf4a8dc3dd4fa9f859b"), "Rollno" : 1, "name" : "nikita", "subject" : "DMSA", "marks" : 78 }

//USING MINIMUM INDEX:

> db.student.find().max({Rollno:3});

{ "\_id" : ObjectId("541c0cf4a8dc3dd4fa9f859b"), "Rollno" : 1, "name" : "nikita", "subject" : "DMSA", "marks" : 78 }

{ "\_id" : ObjectId("541c0d92a8dc3dd4fa9f859c"), "Rollno" : 2, "name" : "kiran", "subject" : "OSD", "marks" : 75 }

//USING MINIMUM INDEX:

> db.student.find().min({Rollno:2});

{ "\_id" : ObjectId("541c0d92a8dc3dd4fa9f859c"), "Rollno" : 2, "name" : "kiran", "subject" : "OSD", "marks" : 75 }

{ "\_id" : ObjectId("541c0ddca8dc3dd4fa9f859d"), "Rollno" : 3, "name" : "komal", "subject" : "TOC", "marks" : 69 }

{ "\_id" : ObjectId("541c0e05a8dc3dd4fa9f859e"), "Rollno" : 4, "name" : "snehal", "subject" : "TOC", "marks" : 81 }

{ "\_id" : ObjectId("541c0e05a8dc3dd4fa9f859f"), "Rollno" : 5, "name" : "saurabh", "subject" : "TOC", "marks" : 78 }

//SORT INDEX IN ASCENDING

> db.student.find().sort({Rollno:1}).pretty();

{

"\_id" : ObjectId("541c0cf4a8dc3dd4fa9f859b"),

"Rollno" : 1,

"name" : "nikita",

"subject" : "DMSA",

"marks" : 78

}

{

"\_id" : ObjectId("541c0d92a8dc3dd4fa9f859c"),

"Rollno" : 2,

"name" : "kiran",

"subject" : "OSD",

"marks" : 75

}

{

"\_id" : ObjectId("541c0ddca8dc3dd4fa9f859d"),

"Rollno" : 3,

"name" : "komal",

"subject" : "TOC",

"marks" : 69

}

{

"\_id" : ObjectId("541c0e05a8dc3dd4fa9f859e"),

"Rollno" : 4,

"name" : "snehal",

"subject" : "TOC",

"marks" : 81

}

{

"\_id" : ObjectId("541c0e2ca8dc3dd4fa9f85a1"),

"Rollno" : 5,

"name" : "saurabh",

"subject" : "TOC",

"marks" : 78

}

//SORTING IN DESCENDING

> db.student.find().sort({Rollno:-1}).pretty();

{

"\_id" : ObjectId("541c0e05a8dc3dd4fa9f859f"),

"Rollno" : 5,

"name" : "saurabh",

"subject" : "TOC",

"marks" : 78

}

{

"\_id" : ObjectId("541c0e2ca8dc3dd4fa9f85a0"),

"Rollno" : 4,

"name" : "snehal",

"subject" : "TOC",

"marks" : 81

}

{

"\_id" : ObjectId("541c0ddca8dc3dd4fa9f859d"),

"Rollno" : 3,

"name" : "komal",

"subject" : "TOC",

"marks" : 69

}

{

"\_id" : ObjectId("541c0d92a8dc3dd4fa9f859c"),

"Rollno" : 2,

"name" : "kiran",

"subject" : "OSD",

"marks" : 75

}

{

"\_id" : ObjectId("541c0cf4a8dc3dd4fa9f859b"),

"Rollno" : 1,

"name" : "nikita",

"subject" : "DMSA",

"marks" : 78

}

//ENSURE INDEX:

> db.student.ensureIndex({Rollno:-1});

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 3,

"numIndexesAfter" : 4,

"ok" : 1

}

> db.student.ensureIndex({Rollno:1});

{ "numIndexesBefore" : 4, "note" : "all indexes already exist", "ok" : 1 }

//GET SIZE OF INDEX:

> db.student.stats();

{

"ns" : "mydb.student",

"count" : 7,

"size" : 784,

"avgObjSize" : 112,

"storageSize" : 8192,

"numExtents" : 1,

"nindexes" : 4,

"lastExtentSize" : 8192,

"paddingFactor" : 1,

"systemFlags" : 1,

"userFlags" : 1,

"totalIndexSize" : 32704,

"indexSizes" : {

"\_id\_" : 8176,

"marks\_1" : 8176,

"Rollno\_1" : 8176,

"Rollno\_-1" : 8176

},

"ok" : 1

}

//EXPLAIN:

> db.student.find().explain();

{

"cursor" : "BasicCursor",

"isMultiKey" : false,

"n" : 7,

"nscannedObjects" : 7,

"nscanned" : 7,

"nscannedObjectsAllPlans" : 7,

"nscannedAllPlans" : 7,

"scanAndOrder" : false,

"indexOnly" : false,

"nYields" : 0,

"nChunkSkips" : 0,

"millis" : 0,

"server" : "gescoe:27017",

"filterSet" : false

}

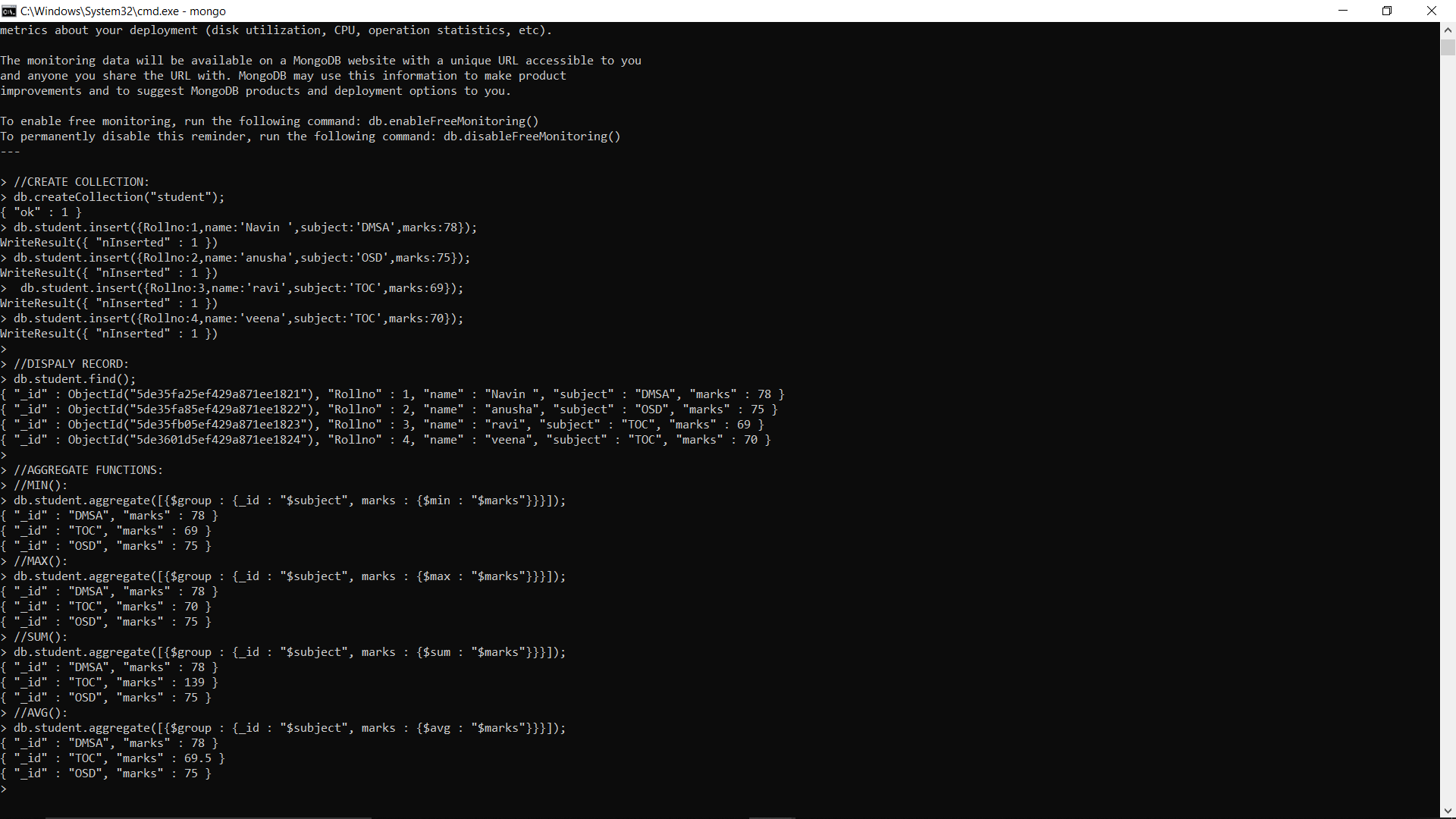
//DROP INDEX:

> db.student.dropIndex({Rollno:1});

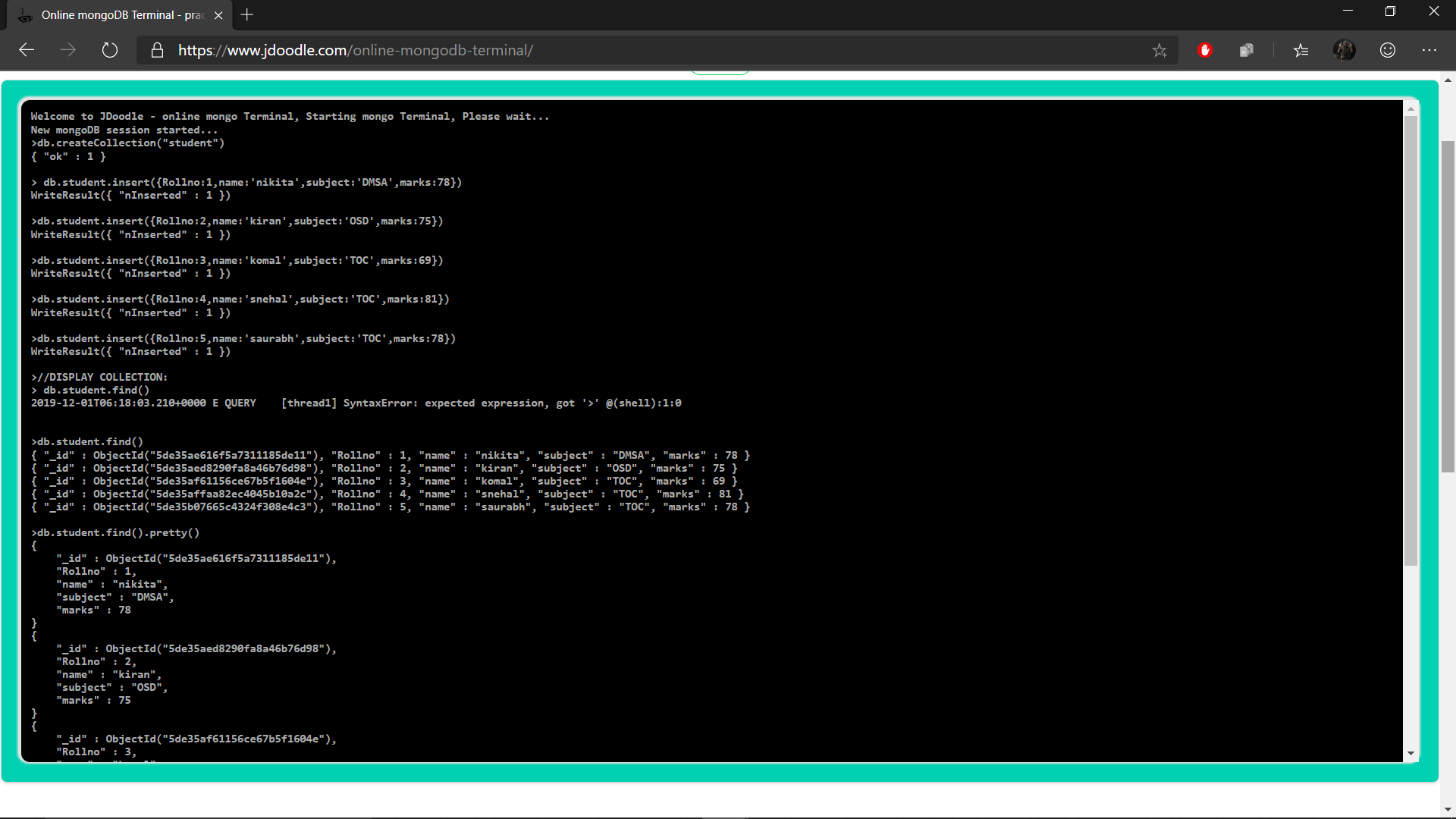
{ "nIndexesWas" : 4, "ok" : 1 }

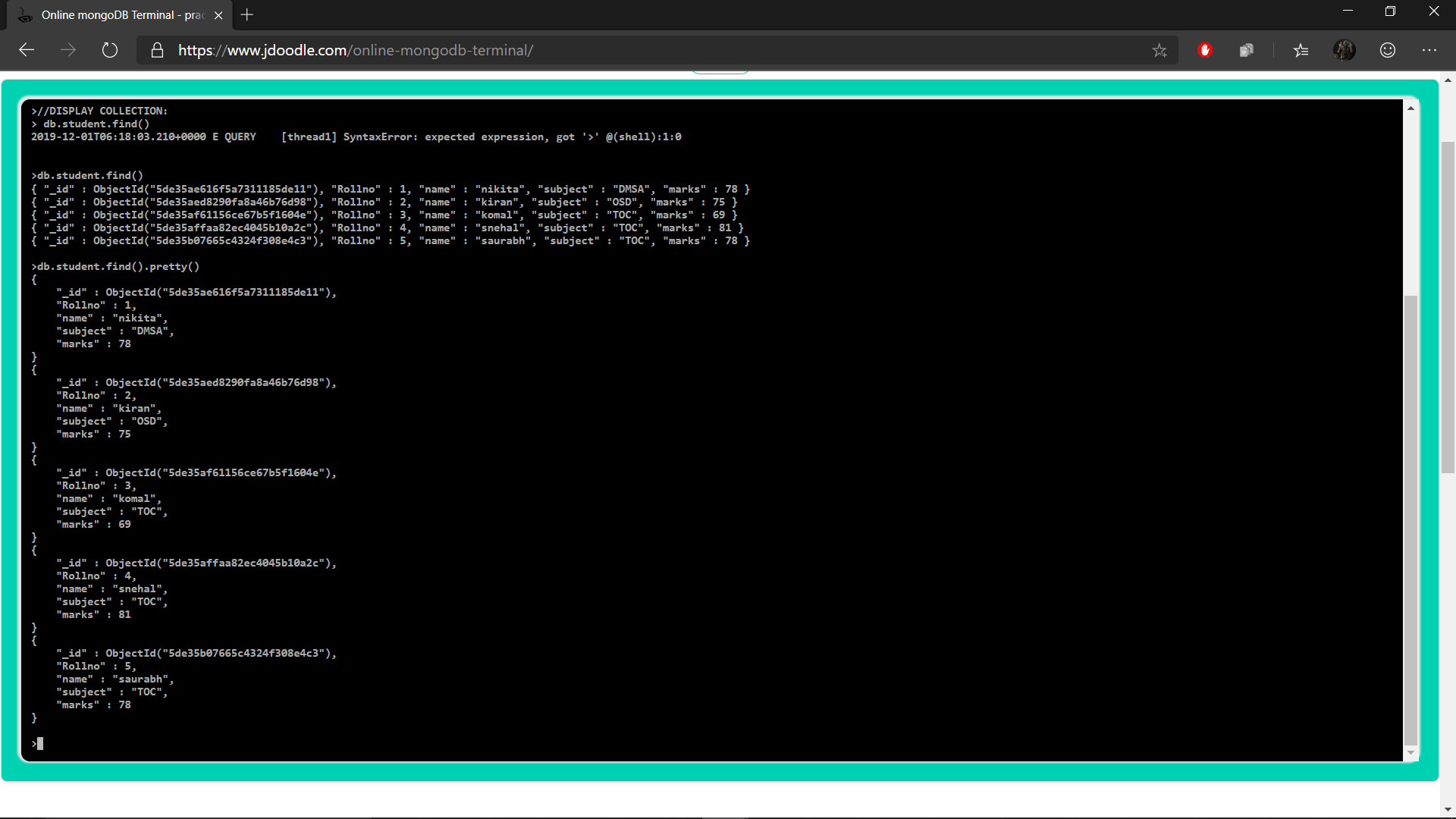
**O/P :**

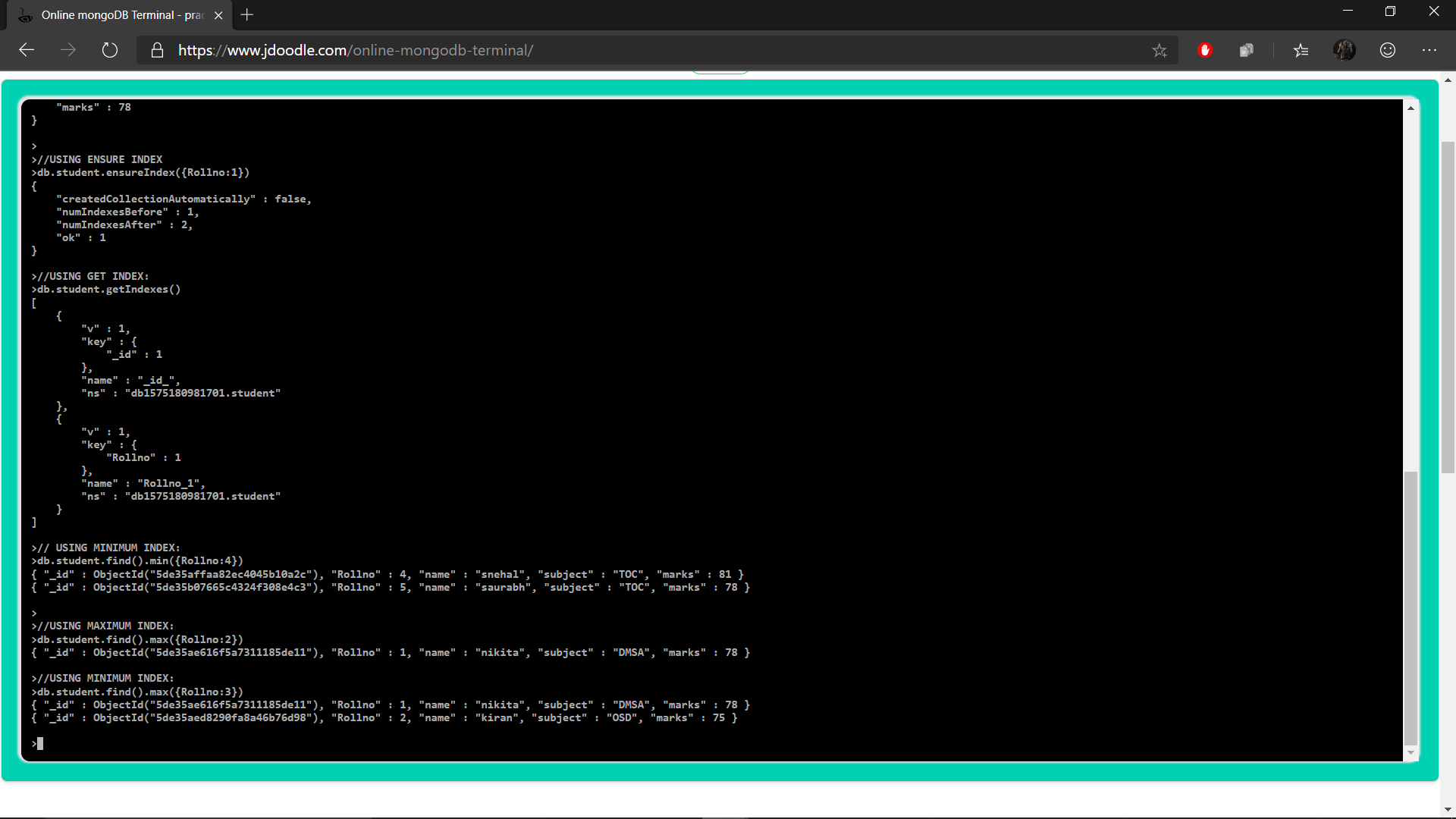
**AGGREGATION:**

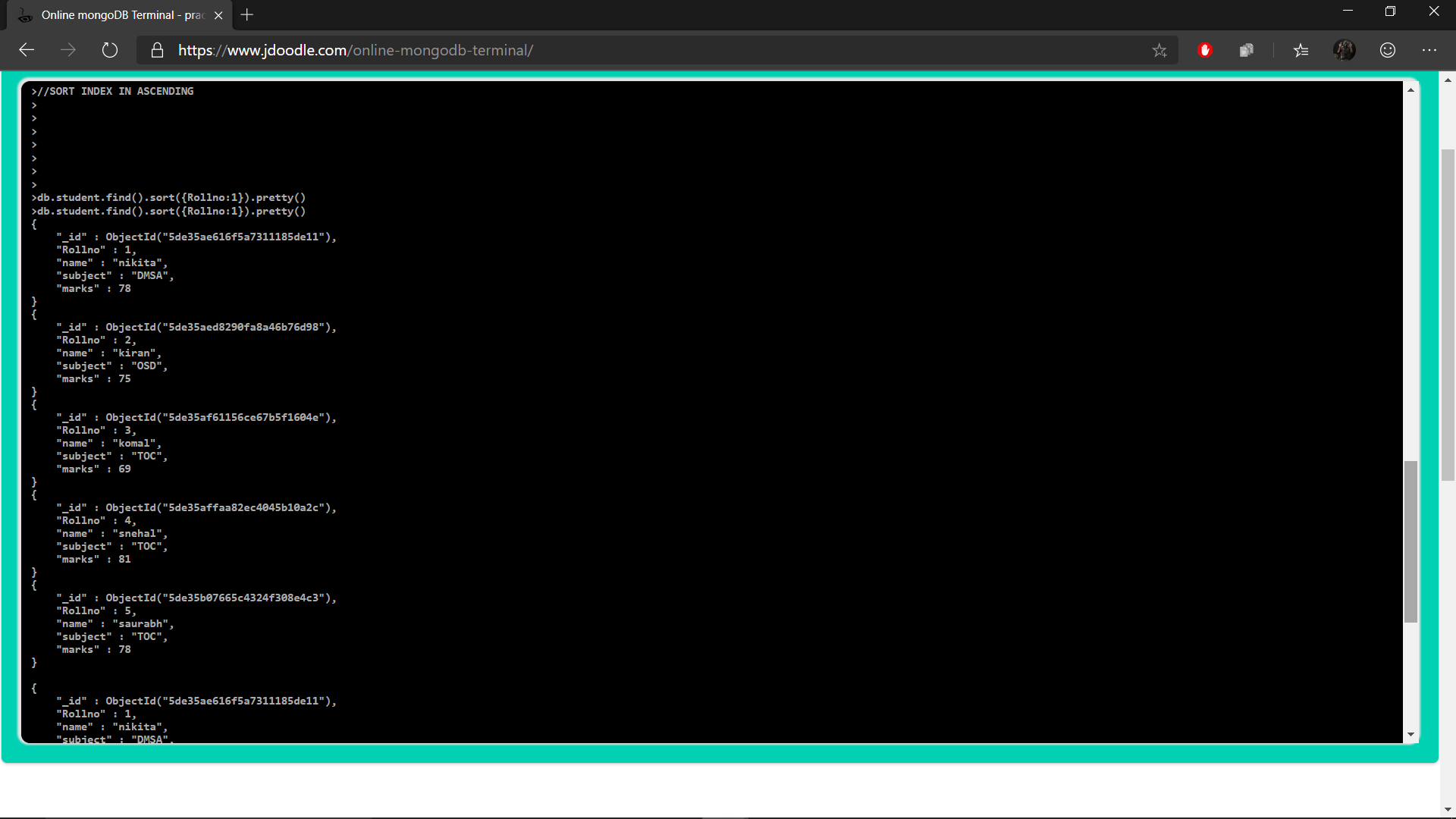
****

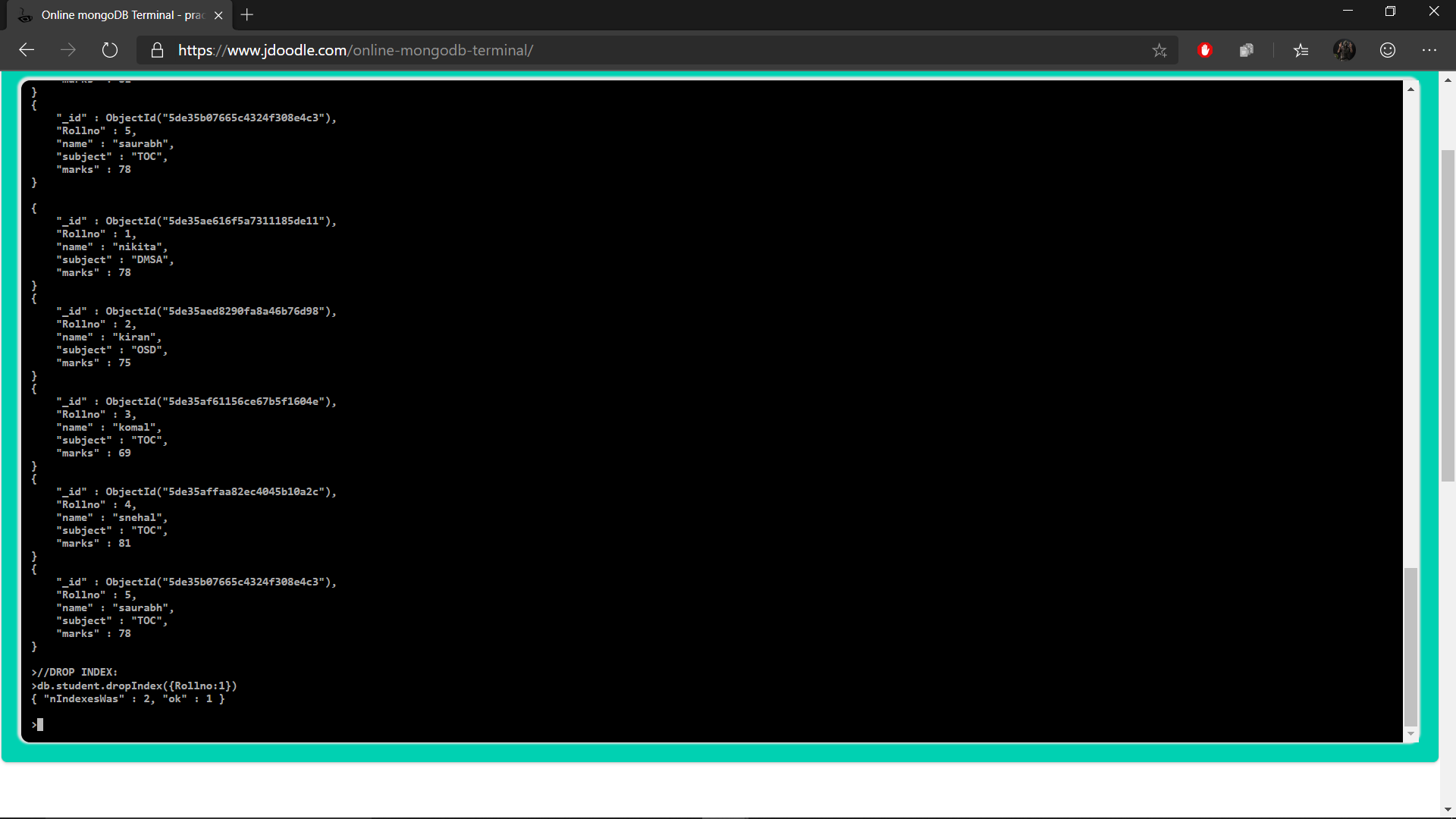
**INDEXES:**

****

****

****

****

****

**CONCLUSION :** Thus we have studied and successfully implemented MongoDB aggregation and indexes

**REMARK :**

|  |  |  |
| --- | --- | --- |
| SIGN | DATE | REMARK |
|  |  |  |