

Ex.No.: 14	MONGO DB
Date:	

MongoDB is a free and open-source cross-platform document-oriented database. Classified as a NoSQL database, MongoDB avoids the traditional table-based relational database structure in favor of JSON-like documents with dynamic schemas, making the integration of data in certain types of applications easier and faster.

Create Database using mongosh

After connecting to your database using mongosh, you can see which database you are using by typing db in your terminal.

If you have used the connection string provided from the MongoDB Atlas dashboard, you should be connected to the myFirstDatabase database.

Show all databases

To see all available databases, in your terminal type show dbs.

Notice that myFirstDatabase is not listed. This is because the database is empty. An empty database is essentially non-existent.

Change or Create a Database

You can change or create a new database by typing use then the name of the database.

Create Collection using mongosh

You can create a collection using the createCollection() database method.

Insert Documents

insertOne()

db.posts.insertOne({

title: "Post Title 1",

body: "Body of post.",

category: "News",

likes: 1,

tags: ["news", "events"],

```
date: Date()
```

```
})
```

### EXERCISE 18

Structure of 'restaurants' collection:

```
{
  "address": {
    "building": "1007",
    "coord": [ -73.856077, 40.848447 ],
    "street": "Morris Park Ave",
    "zipcode": "10462"
  },
  "borough": "Bronx",
  "cuisine": "Bakery",

  "grades": [
    { "date": { "$date": 1393804800000 }, "grade": "A", "score": 2 },
    { "date": { "$date": 1378857600000 }, "grade": "A", "score": 6 },
    { "date": { "$date": 1358985600000 }, "grade": "A", "score": 10 },
    { "date": { "$date": 1322006400000 }, "grade": "A", "score": 9 },
    { "date": { "$date": 1299715200000 }, "grade": "B", "score": 14 }
  ],
  "name": "Morris Park Bake Shop",
  "restaurant_id": "30075445"
}
```

1. Write a MongoDB query to find the restaurant Id, name, borough and cuisine for those restaurants which prepared dish except 'American' and 'Chinees' or restaurant's name begins with letter 'Wil'.
2. Write a MongoDB query to find the restaurant Id, name, and grades for those restaurants which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00.00Z" among many of survey dates..

①) db.restaurants.find({\$or: [{cuisine: '\$\$cin': ['America',  
\$name: '\$\$regno: /us/i?y?]}],

{restaurant\_id: 1, name: 1, borough: 1 cuisine: 1})

②) db.restaurants.find({\$grades: {\$elemMatch \$grade: 'A' score: 11  
date: ISOdate('2014-08-11T00:00:00Z')}?y?  
\$restaurant\_id: 1, name: 1, grades: 1})

3. db.restaurants.find({grades: 1: \$grade A score: 9 date  
ISOdate('2014-08-11T00:00:00Z')}y?

4. db.restaurants.find({\$address.coord: 1: \$gt: 42: \$lte: 52}

5. db.restaurants.find() sort({name: 1})

6. db.restaurants.find().sort({cuisine: 1 borough: -1})

7. db.restaurants.find().sort({cuisine: 1, borough: -1});

8) db.restaurants.find({\$address.coord: {\$type: 'double'?y?});

9. db.restaurants.find({\$grades: {\$elemMatch: \$score: \$  
\$mod: [7,0]?y?}

{restaurant\_id: 1, name: 1 grade: 1}) ;

10. db.restaurants.find({\$grades: {\$all: 2? \$elemMatch:

{score: 2?y? \$elemMatch: \$score: 6?y?}, borough: 1, manhattan: 1})

11. db.restaurants.find({name: {\$regex: /man/i?y?

{name: 1, borough: 1: address.coord: 1 cuisine: 1})

12. db.restaurants.find({name: {\$regex: /mod/i?y?

{name: 1 borough: 1 address.coord: 1 cuisine: 1}) ;

13. db.restaurants.find({\$grades: \$score: \$gt: 5?y?}) ;



14. db. restaurants.find({ 'grades score' : \$lt: 5 }, borough : \$in : ['mahatta'] ) ;
15. db. restaurants.find({ 'grades score' : \$lt: 5 , borough : \$in : ['mahatta'] }
16. db. restaurants.find({ 'grades score' : \$lt: 5 }  
borough : \$in : ['mahatta', 'Brooklyn'] }  
cuisine : \$in : ['Mexican'] } ) ;
17. db. restaurants.find({ 'grades score' : \$lt: 5 }  
borough : \$in : ['mahatta', 'Brooklyn'] }  
cuisine : \$in : ['chinese'] } )
18. db. restaurants.find({ grades : \$all : [ \$elematch : score : 2  
\$elematch : [score : 6] ] }, borough : "mahattan" ) ;
19. db. restaurants.find({ grades : \$all : [ \$elematch :  
\$score : 2 ], \$elematch : [score : 6] ] }, borough : 'manhattan' ) ;
20. db. restaurants.find({ grades : \$all : [ \$elematch :  
[score : 2] , \$elematch : \$score : 6 ] }
21. db. restaurants.find({ grades : \$all : [ \$elematch : \$score : 2  
\$elematch : \$score : 6 ] }  
borough : \$in : ['manhattan', 'Brooklyn'] },  
cuisine : \$in : ['America', 'chinese'] } ) ;
22. db. restaurants.find({ \$or : [ { 'grades score' : 2 ,  
'grades score' : 6 } ] } ) ;