

```
1 DBMSL Assignment No 9
2 Name:Himanshu Vijay Lonkar
3 Batch:T3
4
```

```
5 Output:
6 #Start:
```

```
7
8 test> show dbs
9 admin 40.00 KiB
10 config 12.00 KiB
11 local 40.00 KiB
12
13 test> use Articles
14 switched to db Articles
15 Articles> db.createCollection('articles')
16 { ok: 1 }
17
18 Articles> show dbs
19 Articles 8.00 KiB
20 admin 40.00 KiB
21 config 60.00 KiB
22 local 40.00 KiB
23
```

```
24 1.Insert 5 Data:
```

```
25
26 Articles>
27 db.articles.insertOne({Title:'Entrepreneur',Content:'Businesses',Author:'Bhavik
28 Sarkhedi',Author_age:20,Dop:new
29 Date('2023-10-08'),Category:'FreeLancing',Comments:[{Name:'XYZ',Remarks:'Good'},{
30 Name:'ABC',Remarks:'Can be Better'}]})
31 {
32   acknowledged: true,
33   insertedId: ObjectId("6527c74865659a592e3e2814")
34 }
35
36 Articles> db.articles.insertMany([{Title:'Novelist',Content:'English
37 Literature',Author:'Chetan Bhagat',Author_age:25,Dop:new
38 Date('2023-02-10'),Category:'FreeLancing',Comments:[{Name:'LMN',Remarks:'Good'},{
39 Name:'PQR',Remarks:'Can be Better'}]},{Title:'Story
40 Books',Content:'Horror',Author:'A.B.Sharma',Author_age:30,Dop:new
41 Date('2023-3-15'),Category:'FreeLancing',Comments:[{Name:'AAA',Remarks:'Good'},{N
42 ame:'BBB',Remarks:'Can be Better'}]})
43 {
44   acknowledged: true,
45   insertedIds: {
46     '0': ObjectId("6527c9dc65659a592e3e2815"),
47     '1': ObjectId("6527c9dc65659a592e3e2816")
48   }
49 }
50
51 Articles> db.articles.insertMany([{Title:'Novelist',Content:'Indian
52 Literature',Author:'Anita Desai',Author_age:16,Dop:new
53 Date('2023-05-22'),Category:'FreeLancing',Comments:[{Name:'QQQ',Remarks:'Good'},{
54 Name:'PPP',Remarks:'Can be Better'}]},{Title:'The God of Small
55 Things',Content:'ScreenPlays',Author:'Arundhati Roy',Author_age:16,Dop:new
56 Date('2022-4-20'),Category:'FreeLancing',Comments:[{Name:'DDD',Remarks:'Good'},{N
57 ame:'CCC',Remarks:'Can be Better'}]})
58 {
59   acknowledged: true,
60   insertedIds: {
61     '0': ObjectId("6527cae665659a592e3e2817"),
62     '1': ObjectId("6527cae665659a592e3e2818")
63   }
64 }
```

```
65 1.2 Display all records:
```

```
66 Articles> db.articles.find().pretty()
67 [
```

```

54     {
55         _id: ObjectId("6527c74865659a592e3e2814"),
56         Title: 'Entrepreneur',
57         Content: 'Businesses',
58         Author: 'Bhavik Sarkhedi',
59         Author_age: 20,
60         Dop: ISODate("2023-10-08T00:00:00.000Z"),
61         Category: 'FreeLancing',
62         Comments: [
63             { Name: 'XYZ', Remarks: 'Good' },
64             { Name: 'ABC', Remarks: 'Can be Better' }
65         ]
66     },
67     {
68         _id: ObjectId("6527c9dc65659a592e3e2815"),
69         Title: 'Novelist',
70         Content: 'English Literature',
71         Author: 'Chetan Bhagat',
72         Author_age: 25,
73         Dop: ISODate("2023-02-10T00:00:00.000Z"),
74         Category: 'FreeLancing',
75         Comments: [
76             { Name: 'LMN', Remarks: 'Good' },
77             { Name: 'PQR', Remarks: 'Can be Better' }
78         ]
79     },
80     {
81         _id: ObjectId("6527c9dc65659a592e3e2816"),
82         Title: 'Story Books',
83         Content: 'Horror',
84         Author: 'A.B.Sharma',
85         Author_age: 30,
86         Dop: ISODate("2023-03-14T18:30:00.000Z"),
87         Category: 'FreeLancing',
88         Comments: [
89             { Name: 'AAA', Remarks: 'Good' },
90             { Name: 'BBB', Remarks: 'Can be Better' }
91         ]
92     },
93     {
94         _id: ObjectId("6527cae665659a592e3e2817"),
95         Title: 'Novelist',
96         Content: 'Indian Literature',
97         Author: 'Anita Desai',
98         Author_age: 16,
99         Dop: ISODate("2023-05-22T00:00:00.000Z"),
100        Category: 'FreeLancing',
101        Comments: [
102            { Name: 'QQQ', Remarks: 'Good' },
103            { Name: 'PPP', Remarks: 'Can be Better' }
104        ]
105    },
106    {
107        _id: ObjectId("6527cae665659a592e3e2818"),
108        Title: 'The God of Small Things',
109        Content: 'ScreenPlays',
110        Author: 'Arundhati Roy',
111        Author_age: 16,
112        Dop: ISODate("2022-04-19T18:30:00.000Z"),
113        Category: 'FreeLancing',
114        Comments: [
115            { Name: 'DDD', Remarks: 'Good' },
116            { Name: 'CCC', Remarks: 'Can be Better' }
117        ]
118    }
119 ]
120

```

2.Display first document found in db:

```

Articles> db.articles.find().limit(1)
[
  {
    _id: ObjectId("6527c74865659a592e3e2814"),
    Title: 'Entrepreneur',
    Content: 'Businesses',
    Author: 'Bhavik Sarkhedi',
    Author_age: 20,
    Dop: ISODate("2023-10-08T00:00:00.000Z"),
    Category: 'FreeLancing',
    Comments: [
      { Name: 'XYZ', Remarks: 'Good' },
      { Name: 'ABC', Remarks: 'Can be Better' }
    ]
  }
]

```

3.Display first document belonging to a certain author says "Sharma":

```

Articles> db.articles.find({Author:/Sharma/})
[
  {
    _id: ObjectId("6527c9dc65659a592e3e2816"),
    Title: 'Story Books',
    Content: 'Horror',
    Author: 'A.B.Sharma',
    Author_age: 30,
    Dop: ISODate("2023-03-14T18:30:00.000Z"),
    Category: 'FreeLancing',
    Comments: [
      { Name: 'AAA', Remarks: 'Good' },
      { Name: 'BBB', Remarks: 'Can be Better' }
    ]
  }
]

```

4.Modify the comment by certain person on a certain article:

```

Articles>
db.articles.update({Title:'Entrepreneur','Comments.Name':'ABC'},{$set: {'Comments.
$.Remarks':'Excellent'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Articles> db.articles.find().limit(1)
[
  {
    _id: ObjectId("6527c74865659a592e3e2814"),
    Title: 'Entrepreneur',
    Content: 'Businesses',
    Author: 'Bhavik Sarkhedi',
    Author_age: 20,
    Dop: ISODate("2023-10-08T00:00:00.000Z"),
    Category: 'FreeLancing',
    Comments: [
      { Name: 'XYZ', Remarks: 'Good' },
      { Name: 'ABC', Remarks: 'Excellent' }
    ]
  }
]

```

5.1 Insert records with save method with object \_id:

```

Articles>
db.articles.save({_id:10,Title:'DBMS',Content:'MongoDB',Author:'Prof.M.S.Agarwal'

```

```
,Author_age:30,Dop:new
Date('2023-10-10'),Category:'Database',Comments:[{Name:'HHH',Remarks:'Better'},{Name:'JJJ',Remarks:'Excellent'}]);
```

```
189
190         > db.articles.find({Title:"DBMS"}).pretty()
191
192         {
193           "_id" : 10,
194           "Title" : "DBMS",
195           "Content" : "MongoDB",
196           "Author" : "Prof.M.S.Agarwal",
197           "Author_age" : 30,
198           "Dop" : ISODate("2023-10-10T00:00:00Z"),
199           "Category" : "Database",
200           "Comments" : [
201             {
202               "Name" : "HHH",
203               "Remarks" : "Better"
204             },
205             {
206               "Name" : "JJJ",
207               "Remarks" : "Excellent"
208             },
209             {
210               "Name" : "FFF",
211               "Remarks" : "Nice"
212             }
213           ]
214         }
215
```

5.2 Insert records with save method without object \_id:

```
>
db.articles.save({Title:'TOC',Content:'DFA',Author:'Prof.R.R.Bathe',Author_age:30,
,Dop:new
Date('2023-10-20'),Category:'Automata',Comments:[{Name:'EEE',Remarks:'Better'},
{Name:'SSS',Remarks:'Excellent'}]});
WriteResult({ "nInserted" : 1 })
219
220
221 > db.articles.find({Title:"TOC"}).pretty()
222 {
223   "_id" : ObjectId("652cd16e3110241d64b6456d"),
224   "Title" : "TOC",
225   "Content" : "DFA",
226   "Author" : "Prof.R.R.Bathe",
227   "Author_age" : 30,
228   "Dop" : ISODate("2023-10-20T00:00:00Z"),
229   "Category" : "Automata",
230   "Comments" : [
231     {
232       "Name" : "EEE",
233       "Remarks" : "Better"
234     },
235     {
236       "Name" : "SSS",
237       "Remarks" : "Excellent"
238     }
239   ]
240 }
241
242
```

6.Update a Collection with save method:

```
>
db.articles.save({_id:10,Title:'DBMS',Content:'MySQL',Author:'Prof.M.S.Agarwal',Author_age:25,Dop:new
Date("2023-10-16"),Category:'Database',Comments:[{Name:'HHH',Remarks:'Good'},
{Name:'JJJ',Remarks:'Nice'}]});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
246
```

```

247
248 > db.articles.find({Title:'DBMS'}).pretty()
249 {
250     "_id" : 10,
251     "Title" : "DBMS",
252     "Content" : "MySQL",
253     "Author" : "Prof.M.S.Agarwal",
254     "Author_age" : 25,
255     "Dop" : ISODate("2023-10-16T00:00:00Z"),
256     "Category" : "Database",
257     "Comments" : [
258         {
259             "Name" : "HHH",
260             "Remarks" : "Good"
261         },
262         {
263             "Name" : "JJJ",
264             "Remarks" : "Nice"
265         }
266     ]
267 }
268
269
270 7.Add one more comment for a particular Title:
271
272 >
273 db.articles.update({Title:'TOC'},{$push:{Comments:{Name:'FFF',Remarks:'Nice'}}})
274 WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
275
276 > db.articles.find({Title:'TOC'}).pretty()
277 {
278     "_id" : ObjectId("652cd16e3110241d64b6456d"),
279     "Title" : "TOC",
280     "Content" : "DFA",
281     "Author" : "Prof.R.R.Bathe",
282     "Author_age" : 30,
283     "Dop" : ISODate("2023-10-20T00:00:00Z"),
284     "Category" : "Automata",
285     "Comments" : [
286         {
287             "Name" : "EEE",
288             "Remarks" : "Better"
289         },
290         {
291             "Name" : "SSS",
292             "Remarks" : "Excellent"
293         },
294         {
295             "Name" : "FFF",
296             "Remarks" : "Nice"
297         }
298     ]
299 }
300
301 8.Delete the documents whose author age<18:
302
303 > db.articles.deleteMany({Author_age:{$lt:18}})
304 { "acknowledged" : true, "deletedCount" : 2 }
305
306 > db.articles.find().pretty()
307 {
308     "_id" : ObjectId("6528cd46a0571873b9d71186"),
309     "Title" : "Entrepreneur",
310     "Content" : "Businesses",
311     "Author" : "Bhavik Sarkhedi",
312     "Author_age" : 20,
313     "Dop" : ISODate("2023-10-08T00:00:00Z"),
314     "Category" : "FreeLancing",

```

```

315         "Comments" : [
316             {
317                 "Name" : "XYZ",
318                 "Remarks" : "Good"
319             },
320             {
321                 "Name" : "ABC",
322                 "Remarks" : "Excellent"
323             }
324         ]
325     }
326 {
327     "_id" : ObjectId("6528cee6a0571873b9d71187"),
328     "Title" : "Novelist",
329     "Content" : "English Literature",
330     "Author" : "Chetan Bhagat",
331     "Author_age" : 25,
332     "Dop" : ISODate("2023-02-10T00:00:00Z"),
333     "Category" : "FreeLancing",
334     "Comments" : [
335         {
336             "Name" : "LMN",
337             "Remarks" : "Good"
338         },
339         {
340             "Name" : "PQR",
341             "Remarks" : "Can be Better"
342         }
343     ]
344 }
345 {
346     "_id" : ObjectId("6528cee6a0571873b9d71188"),
347     "Title" : "Story Books",
348     "Content" : "Horror",
349     "Author" : "A.B.Sharma",
350     "Author_age" : 30,
351     "Dop" : ISODate("2023-03-15T00:00:00Z"),
352     "Category" : "FreeLancing",
353     "Comments" : [
354         {
355             "Name" : "AAA",
356             "Remarks" : "Good"
357         },
358         {
359             "Name" : "BBB",
360             "Remarks" : "Can be Better"
361         }
362     ]
363 }
364 {
365     "_id" : 10,
366     "Title" : "DBMS",
367     "Content" : "MySql",
368     "Author" : "Prof.M.S.Agarwal",
369     "Author_age" : 25,
370     "Dop" : ISODate("2023-10-16T00:00:00Z"),
371     "Category" : "Database",
372     "Comments" : [
373         {
374             "Name" : "HHH",
375             "Remarks" : "Good"
376         },
377         {
378             "Name" : "JJJ",
379             "Remarks" : "Nice"
380         }
381     ]
382 }
383 {

```

```
384         "_id" : ObjectId("652cd16e3110241d64b6456d"),
385         "Title" : "TOC",
386         "Content" : "DFA",
387         "Author" : "Prof.R.R.Bathe",
388         "Author_age" : 30,
389         "Dop" : ISODate("2023-10-20T00:00:00Z"),
390         "Category" : "Automata",
391         "Comments" : [
392             {
393                 "Name" : "EEE",
394                 "Remarks" : "Better"
395             },
396             {
397                 "Name" : "SSS",
398                 "Remarks" : "Excellent"
399             },
400             {
401                 "Name" : "FFF",
402                 "Remarks" : "Nice"
403             }
404         ]
405     }
406
407
```

408 9.Remove all the documents from the collection:

```
409
410     > db.articles.deleteMany({})
411     { "acknowledged" : true, "deletedCount" : 5 }
412
413     > db.articles.find().pretty()
414
```

415 10.Delete article collection:

```
416
417     > db.articles.drop()
418     true
419
420     > db.getCollectionNames()
421     [ ]
422
423
424
```