



Create an empty collection named "student"

- db.student.insert({

regno: "3011",

name: "Test Student",

CourseName: "MCA",
duration: 2,

gradPer: 70,

address: {

city: "Changa",

state: "GJ",

country: "India",

}

- db.student.insert({

regNo: "3015",

name: "Jhanvi",

CourseName: "BCA",

duration: 3,

HSCPer: 80,

address: {

city: "Saurat",

state: "Guj",

country: "India",

}

)

Insert at least 5 documents.

1. change the name to "Ami" for Test document.

=> db.student.update({ "name": "Test Student" }, { \$set: { "name": "Ami" } })

2. Find student details whose name starts with "A"

=> db.student.find({ "name": /^A/ })

3. Find student details whose name ends with "i"

=> db.student.find({ "name": /i\$/ })

4. Find student details registered for 3 year course.

=> db.student.find({ "duration": 3 })

5. Find student registration number, name for all the students.

=> db.student.find({ }, { "regno": 1, -id: 0, name: 1 })

6. Find student registration number, name, course name and course duration in ascending order by their.

=> db.student.find({ }, { "regno": 1, "name": 1, "courcename": 1, "duration": 1, -id: 0 }).
sort({ duration: 1 })

7. Find total number of students.

=> db.student.find().count()

8. Find total number of student studying in MCA course.

=> db.student.find({ "CourseName": "MCA" }).count()

9. Find average Percentage for MCA and BCA students.

=> db.student.aggregate([
 { \$group: { "-id": "\$CourseName",
 Average: { \$avg: "\$last" } } }
])

=> db.student.updateMany({}, { \$set: { "last": 80 } })

10. Find average Percentage for MCA students.

=> db.student.aggregate([
 { \$match: { CourseName: "MCA" } },
 { \$group: { "-id": "\$CourseName",
 Average: { \$avg: "\$last" } } }
])

11. find number of students studying in MCA with more than 60 Percentage in graduation.

=> db.student.find({ "\$and": [{ "CourseName": "MCA" },
 { "last": { \$gt: 60 } }] }).count()

12. Find average Percentage for each course. Eliminate the course if average percentage is less than 6.

=> db.student.aggregate([

{ \$match: { "last": { \$get: "B0444" } }

{ \$group: { "_id": "\$CourseName", }

Average: { "\$avg": "\$last" } }])

13. find total number of students whose name starts with "A".

=> db.student.find({ "name": /^AA/ }).count()

14. find total number of student who are not from state "Orj".

=> db.student.find({ "address.state": { "\$ne": "Orj" } }).count()

db.student.aggregate([

{

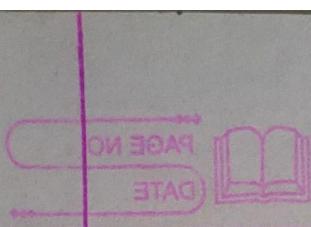
\$group: { _id: "\$courseName" },

AverageMCA: { \$avg: '\$gradPer' },

AverageBCA: { \$avg: '\$itscPer' } }

]

])



ON BOARD

PAGE NO.

DATE

```
    "dmatch": {"$or": [{"AverageBCA": {"$get": 50}}, {"AverageMCA": {"$get": 50}}]}
```

y

1

Since AgNO_3 precipitates to form a white solid.

三

(1) $\text{MgO} \cdot (\text{SiO}_4)$ and its hydrates [L. 1-]

tert Ben oden brachte de gedreven lotst huid.

Untested.

7) tiny.cc/meyarw

the "DRAFTING" - "PRINTING" - "ADVERTISING" -