

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)
DEPARTMENT OF COMPUTER APPLICATIONS

ADVANCED DBMS LAB
(Semester & Batch: S2 MCA 2020 B Batch)

CYCLE -3

NoSQL Database – Designing, Query Processing,

CO 3	Configuration of NoSQL database
CO 4	Apply CRUD operations and retrieve data in a NoSQL environment.

1. Install the MongoDB Compass GUI and configure it.
2. Create a collection student consists of details like rollno, name, phoneno, marks, address, year of course etc

OUTPUT

use student

switched to db student

```
> db.createCollection("stud")
```

```
{  
  "ok" : 0,  
  "errmsg" : "a collection 'student.stud' already exists",  
  "code" : 48,  
  "codeName" : "NamespaceExists"  
}
```

2) Insert the details of the multiple students (atleast 5) in the form of documents in the student collection.

OUTPUT

```
joseph", "mark" : 150, "city" : "kochi", "ph" : 679832, "year" : 2020 }
```

```
{ "_id" : 102, "name" : "joeh", "mark" : 50, "city" : "mumbai", "ph" : 349832, "year" : 2022 }
```

```
{ "_id" : 103, "name" : "joe", "mark" : 100, "city" : "milan", "ph" : 347832, "year" : 2022 }
```

```
{ "_id" : 104, "name" : "joel", "mark" : 120, "city" : "trissur", "ph" : 70347832, "year" : 2021 }
```

```
{ "_id" : 105, "name" : "polo", "mark" : 130, "city" : "aluva", "ph" : 7034376217, "year" : 2010 }
```

```
> db.stud.find().pretty()
```

3. Retrieve the fields rollno, name, phoneno, marks, city for all the documents in the collection student.

OUTPUT

```
> db.stud.find().pretty()
```

```
{
```

```
  "_id" : 101,
```

```
  "name" : "joseph",
```

```
  "mark" : 150,
```

```
  "city" : "kochi",
```

```
  "ph" : 679832,
```

```
  "year" : 2020
```

```
}
```

```
{
```

```
  "_id" : 102,
```

```
  "name" : "joeh",
```

```
  "mark" : 50,
```

```
  "city" : "mumbai",
```

```
  "ph" : 349832,
```

```
  "year" : 2022
```

```
}
```

```
{
```

```
"_id" : 103,  
"name" : "joe",  
"mark" : 100,  
"city" : "milan",  
"ph" : 347832,  
"year" : 2022  
}  
{  
  "_id" : 104,  
  "name" : "joel",  
  "mark" : 120,  
  "city" : "trissur",  
  "ph" : 70347832,  
  "year" : 2021  
}  
{  
  "_id" : 105,  
  "name" : "polo",  
  "mark" : 130,  
  "city" : "aluva",  
  "ph" : 7034376217,  
  "year" : 2010
```

4. Display the details of students who achieved a score more than 90 and are from 'Thrissur'.

OUTPUT

```
> db.stud.find({$and:[{mark:{$gt:90}},{city:"trissur"}]}).pretty()  
{  
  "_id" : 104,  
  "name" : "joel",  
  "mark" : 120,  
  "city" : "trissur",
```

```
"ph" : 70347832,  
"year" : 2021  
}
```

5. Update the phone number of Sujith in the student collection. Retrieve the updated information.

OUTPUT

```
db.stud.updateMany({$set:{year:2021}})  
2021-09-29T16:12:14.862+0530 E QUERY [js] uncaught exception: TypeError:  
can't convert undefined to object :  
DBCollection.prototype.updateMany@src/mongo/shell/crud_api.js:648:20  
@(shell):1:1
```

6. Update the year of course in all the documents in the student collection to 2021. Also retrieve the updated information.

OUTPUT

```
> db.stud.updateMany({year:2020},{set:{year:2021}})  
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }  
> db.stud.find().pretty()  
{  
  "_id" : 101,  
  "name" : "joseph",  
  "mark" : 150,  
  "city" : "kochi",  
  "ph" : 679832,  
  "year" : 2021  
}  
{  
  "_id" : 102,  
  "name" : "joeh",  
  "mark" : 50,  
  "city" : "mumbai",
```

```
"ph" : 349832,  
"year" : 2022  
}  
{  
  "_id" : 103,  
  "name" : "joe",  
  "mark" : 100,  
  "city" : "milan",  
  "ph" : 347832,  
  "year" : 2022  
}  
{  
  "_id" : 104,  
  "name" : "joel",  
  "mark" : 120,  
  "city" : "trissur",  
  "ph" : 7012888132,  
  "year" : 2021  
}  
{  
  "_id" : 105,  
  "name" : "polo",  
  "mark" : 130,  
  "city" : "aluva",  
  "ph" : 7034376217,  
  "year" : 2010  
}
```

7)Delete the details of the student whose name is 'Abhilash' from the student collection

OUTPUT

```
db.stud.deleteOne({name:"joseph"})
```

```
{ "acknowledged" : true, "deletedCount" : 1 }
```

9)Retrieve the number of students per department from the student collection.

OUTPUT

```
> db.stud.deleteOne({name:"joseph"})
```

10)Arrange the name of the students in ascending order along with all the columns.

OUTPUT

```
db.stud.find().sort({_id:-1})
```

```
{ "_id" : 105, "name" : "polo", "mark" : 130, "city" : "aluva", "ph" : 7034376217, "year" :  
  2010 }
```

```
{ "_id" : 104, "name" : "joel", "mark" : 120, "city" : "trissur", "ph" : 7012888132, "year" :  
  2021 }
```

```
{ "_id" : 103, "name" : "joe", "mark" : 100, "city" : "milan", "ph" : 347832, "year" : 2022 }
```

```
{ "_id" : 102, "name" : "joeh", "mark" : 50, "city" : "mumbai", "ph" : 349832, "year" : 2022  
  }
```

```
>
```