DEPARTMENT OF MCA

SEM – 3

BIG DATA ANALYATICS AND VISUALIZATION LAB

PRACTICAL NO 3.

Exercise – I: Student Database

Agenda: Create database, Create collection, insert data, find, find one, sort, limit, skip, distinct, projection.

Create a student database with the fields: (SRN, Sname, Degree, Sem, CGPA)

```
> use student1
switched to db student1

>db.stud1col1.insert({srn:110,sname:"Rahul",degree:"BCA",sem:6,CGPA:7.9})
OR
> doc1=({srn:110,sname:"Rahul",degree:"BCA",sem:6,CGPA:7.9})
>db.studcol1.insert (doc1)
```

Note: insert 10 documents.

Questions:

1.display all the documents

```
>db.studcol1.find()
```

2.Display all the students in BCA

```
>db.studcol1.find({degree:"BCA"})
```

3.Display all the students in ascending order

```
>db. studcol1.find().sort({sname:1})
```

4.Display first 5 students

```
>db. studcol1.find().limit(5)
```

5.display students 5,6,7

```
>db. studcol1.find().skip(4).limit(3)
```

6.list the degree of student "Rahul"

```
>db. studcol1.find({degree:1, sname:"Rahul"})
```

7.Display students details of 5,6,7 in descending order of percentage

```
>db. studcol1.find().sort({CGPA:-1}).skip(4).limit(3)
```

8.Display the number of students in BCA

```
>db. studcol1.find({degree:"BCA"}).count()
```

9.Display all the degrees without _id
>db. studcol1.find({},{_id:0})

10.Display all the distinct degrees
>db. studcol1.distinct("degree")

11.Display all the BCA students with CGPA greater than 6, but less than 7.5
>db. studcol1.find(degree:"BCA", {CGPA:{$gt:6, $lt:7.5}})

12.Display all the students in BCA and in 6th Sem
>db. studcol1.find({$and:[{degree:"BCA"},{sem:6}]})

Exercise – II: Employee Database

Agenda: Update modifiers ($set, $unset, $inc, $push, $pushAll, $pull, $pullAll, $addToSet)

Create an employee database with the fields: {eid, ename, dept, desig, salary, yoj, address{dno, street, locality, city}}

```
> use empdb9
switched to db empdb9

> doc1 = {eid:001, ename:"Rahul", dept:"production", desig:"developer", salary:30000, yoj:2015,
          address:{dno:397, street:2, locality:"rmnagar", city:"bangalore"} }
{
        "eid" : 1,
        "ename" : "Rahul",
        "dept" : "production",
        "desig" : "developer",
        "salary" : 30000,
        "yoj" : 2015,
        "address" : {
        "dno" : 397,
        "street" : 2,
        "locality" : "rmnagar",
        "city" : "bangalore"
        }
}
>db.emp09.insert(doc1)
WriteResult({ "nInserted" : 1 })
```

Note: insert 10 documents.

Questions:

1.Display all the employees with salary in range (50000, 75000)
>db.emp09.find({salary:{$gt:50000, $lt:75000}})

2.Display all the employees with desig developer
>db.emp09.find({desig:"developer"})

3.Display the Salary of “Rahul”
>db.emp09.find({ename:"Rahul"},{salary:1})

4.Display the city of employee “Rahul”
>db.emp09.find({ename:"Rahul"},{"address.city":1})

5.Update the salary of developers by 5000 increment
>db.emp09.update({desig:"developer"},{$inc:{"salary":5000}})

6.Add field age to employee “Rahul”
>db.emp09.update({ename:"Rahul"},{$set:{age:"22"}})

7.Remove YOJ from “Rahul”
>db.emp09.update({ename:"Rahul"},{$unset:{yoj:1}})

8.Add an array field project to “Rahul”
>db.emp09.update({ename:"Rahul"},{$push:{projects:"p1"}})

9.Add p2 and p3 project to “Rahul”
>db.emp09.update({ename:"Rahul"},{$pushAll:{projects:["p2","p3"]}})

10.Remove p3 from “Rahul”
>db.emp09.update({ename:"Rahul"},{$pull:{projects:"p3"}})

11.Add a new embedded object “contacts” with “email” and “phone” as array objects to “Rahul”
>db.emp09.update({ename:"Rahul"},{$push:{contacts:{phone:"9036240380", email:"ra
hulpugal.0308@gmail.com"}}})

12.Add two phone numbers to “Rahul”
>db.emp09.update({ename:"Rahul"},{$addToSet:{"contact.phone":[9738751143,988073
0784]}})