M21DG0206		Course Type				
	NoSQL Lab		**L**	**T**	**P**	**C**
Duration : 26 Hrs		HC	0	0	2	2

LAB EXPERIMENTS:

CRUD Operations in MONGODB

1: Student Database

Create a Student database with the fields: (SRN, sname, degree, sem, CGPA)

i. Insert 10 documents.

o/p

>db.stud.insert({SRN:"R21DG043",sname:"santhosh",degree:"MSc",sem:"1st",CGPA:9.1})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DG099",sname:"ramesh",degree:"MSc",sem:"1st",CGPA:8.1})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DC033",sname:"kiran",degree:"MCA",sem:"1st",CGPA:7.5})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DC034",sname:"rakesh",degree:"MCA",sem:"2st",CGPA:8.1})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DB034",sname:"mohith",degree:"BCA",sem:"1st",CGPA:7.1})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DB039",sname:"rahul",degree:"BCA",sem:"2nd",CGPA:7.9})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DB044",sname:"rohith",degree:"BCA",sem:"5th",CGPA:6.9})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DG067",sname:"vignesh",degree:"MSc",sem:"2nd",CGPA:7.7})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DB033",sname:"mohith",degree:"BCA",sem:"4th",CGPA:9.4})

WriteResult({ "nInserted" : 1 })

> db.stud.insert({SRN:"R21DG067",sname:"ranjith",degree:"MSc",sem:"2nd",CGPA:8.7})

WriteResult({ "nInserted" : 1 })

ii. Display all the documents.

o/p

> db.stud.find()

{ "_id" : ObjectId("62930d144c0403136b43e940"), "SRN" : "R21DG043", "sname" : "santhosh", "degree" : "MSc", "sem" : "1st", "CGPA" : 9.1 }

{ "_id" : ObjectId("62930d4b4c0403136b43e941"), "SRN" : "R21DG099", "sname" : "ramesh", "degree" : "MSc", "sem" : "1st", "CGPA" : 8.1 }

{ "_id" : ObjectId("62930d8c4c0403136b43e942"), "SRN" : "R21DC033", "sname" : "kiran", "degree" : "MCA", "sem" : "1st", "CGPA" : 7.5 }

{ "_id" : ObjectId("62930db54c0403136b43e943"), "SRN" : "R21DC034", "sname" : "rakesh", "degree" : "MCA", "sem" : "2st", "CGPA" : 8.1 }

{ "_id" : ObjectId("62930de14c0403136b43e944"), "SRN" : "R21DB034", "sname" : "mohith", "degree" : "BCA", "sem" : "1st", "CGPA" : 7.1 }

{ "_id" : ObjectId("62930e0c4c0403136b43e945"), "SRN" : "R21DB039", "sname" : "rahul", "degree" : "BCA", "sem" : "2nd", "CGPA" : 7.9 }

{ "_id" : ObjectId("62930e3b4c0403136b43e946"), "SRN" : "R21DB044", "sname" : "rohith", "degree" : "BCA", "sem" : "5th", "CGPA" : 6.9 }

{ "_id" : ObjectId("62930e784c0403136b43e947"), "SRN" : "R21DG067", "sname" : "vignesh", "degree" : "MSc", "sem" : "2nd", "CGPA" : 7.7 }

{ "_id" : ObjectId("62930ea34c0403136b43e948"), "SRN" : "R21DB033", "sname" : "mohith", "degree" : "BCA", "sem" : "4th", "CGPA" : 9.4 }

{ "_id" : ObjectId("62930ed54c0403136b43e949"), "SRN" : "R21DG067", "sname" : "ranjith", "degree" : "MSc", "sem" : "2nd", "CGPA" : 8.7 }

iii. Display all the students in BCA.

o/p

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

{ "_id" : ObjectId("62930de14c0403136b43e944"), "SRN" : "R21DB034", "sname" : "mohith", "degree" : "BCA", "sem" : "1st", "CGPA" : 7.1 }

{ "_id" : ObjectId("62930e0c4c0403136b43e945"), "SRN" : "R21DB039", "sname" : "rahul", "degree" : "BCA", "sem" : "2nd", "CGPA" : 7.9 }

{ "_id" : ObjectId("62930e3b4c0403136b43e946"), "SRN" : "R21DB044", "sname" : "rohith", "degree" : "BCA", "sem" : "5th", "CGPA" : 6.9 }

2

{ "_id" : ObjectId("62930ea34c0403136b43e948"), "SRN" : "R21DB033", "sname" : "mohith", "degree" : "BCA", "sem" : "4th", "CGPA" : 9.4 }

iv. Display all the students in ascending order.

o/p

> db.stud.find({},{sname:1,_id:0}).sort({sname:1})

{ "sname" : "kiran" }

{ "sname" : "mohith" }

{ "sname" : "mohith" }

{ "sname" : "rahul" }

{ "sname" : "rakesh" }

{ "sname" : "ramesh" }

{ "sname" : "ranjith" }

{ "sname" : "rohith" }

{ "sname" : "santhosh" }

{ "sname" : "vignesh" }

v. Display first 5 students.

o/p

> db.stud.find().limit(5)

{ "_id" : ObjectId("62930d144c0403136b43e940"), "SRN" : "R21DG043", "sname" : "santhosh", "degree" : "MSc", "sem" : "1st", "CGPA" : 9.1 }

{ "_id" : ObjectId("62930d4b4c0403136b43e941"), "SRN" : "R21DG099", "sname" : "ramesh", "degree" : "MSc", "sem" : "1st", "CGPA" : 8.1 }

{ "_id" : ObjectId("62930d8c4c0403136b43e942"), "SRN" : "R21DC033", "sname" : "kiran", "degree" : "MCA", "sem" : "1st", "CGPA" : 7.5 }

{ "_id" : ObjectId("62930db54c0403136b43e943"), "SRN" : "R21DC034", "sname" : "rakesh", "degree" : "MCA", "sem" : "2st", "CGPA" : 8.1 }

{ "_id" : ObjectId("62930de14c0403136b43e944"), "SRN" : "R21DB034", "sname" : "mohith", "degree" : "BCA", "sem" : "1st", "CGPA" : 7.1 }

vi. Display students 5,6,7.

o/p

> db.stud.find().skip(4).limit(3)

{ "_id" : ObjectId("62930de14c0403136b43e944"), "SRN" : "R21DB034", "sname" : "mohith", "degree" : "BCA", "sem" : "1st", "CGPA" : 7.1 }

{ "_id" : ObjectId("62930e0c4c0403136b43e945"), "SRN" : "R21DB039", "sname" : "rahul", "degree" : "BCA", "sem" : "2nd", "CGPA" : 7.9 }

{ "_id" : ObjectId("62930e3b4c0403136b43e946"), "SRN" : "R21DB044", "sname" : "rohith", "degree" : "BCA", "sem" : "5th", "CGPA" : 6.9 }

vii. List the degree of student "Rahul".

o/p

> db.stud.find({sname:"rahul"},{degree:1,sname:1,_id:0})

{ "sname" : "rahul", "degree" : "BCA" }

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

o/p

> db.stud.find().skip(4).limit(3).sort({CGP:-1})

{ "_id" : ObjectId("62930e0c4c0403136b43e945"), "SRN" : "R21DB039", "sname" : "rahul", "degree" : "BCA", "sem" : "2nd", "CGPA" : 7.9 }

{ "_id" : ObjectId("62930d8c4c0403136b43e942"), "SRN" : "R21DC033", "sname" : "kiran", "degree" : "MCA", "sem" : "1st", "CGPA" : 7.5 }

{ "_id" : ObjectId("62930e3b4c0403136b43e946"), "SRN" : "R21DB044", "sname" : "rohith", "degree" : "BCA", "sem" : "5th", "CGPA" : 6.9 }

ix. Display the number of students in BCA.

o/p

> db.stud.count({degree:"BCA"})

4

x. Display all the degrees without _id.

o/p

> db.stud.find({},{_id:0})

{ "SRN" : "R21DG043", "sname" : "santhosh", "degree" : "MSc", "sem" : "1st", "CGPA" : 9.1 }

{ "SRN" : "R21DG099", "sname" : "ramesh", "degree" : "MSc", "sem" : "1st", "CGPA" : 8.1 }

{ "SRN" : "R21DC033", "sname" : "kiran", "degree" : "MCA", "sem" : "1st", "CGPA" : 7.5 }

{ "SRN" : "R21DC034", "sname" : "rakesh", "degree" : "MCA", "sem" : "2st", "CGPA" : 8.1 }

{ "SRN" : "R21DB034", "sname" : "mohith", "degree" : "BCA", "sem" : "1st", "CGPA" : 7.1 }

{ "SRN" : "R21DB039", "sname" : "rahul", "degree" : "BCA", "sem" : "2nd", "CGPA" : 7.9 }

{ "SRN" : "R21DB044", "sname" : "rohith", "degree" : "BCA", "sem" : "5th", "CGPA" : 6.9 }

{ "SRN" : "R21DG067", "sname" : "vignesh", "degree" : "MSc", "sem" : "2nd", "CGPA" : 7.7 }

{ "SRN" : "R21DB033", "sname" : "mohith", "degree" : "BCA", "sem" : "4th", "CGPA" : 9.4 }

{ "SRN" : "R21DG067", "sname" : "ranjith", "degree" : "MSc", "sem" : "2nd", "CGPA" : 8.7 }

xi. Display all the distinct degrees.

o/p

> db.stud.distinct("degree")

["BCA", "MCA", "MSc"]

xii. Display all the BCA students with CGPA greater than 6, but less than 9.

o/p

> db.stud.find({$and:[{degree:"BCA"},{CGPA:{$gt:6}},{CGPA:{$lt:9}}]}).pretty()

```
{
     "_id" : ObjectId("62930de14c0403136b43e944"),
     "SRN" : "R21DB034",
     "sname" : "mohith",
     "degree" : "BCA",
     "sem" : "1st",
     "CGPA" : 7.1
}
{
     "_id" : ObjectId("62930e0c4c0403136b43e945"),
     "SRN" : "R21DB039",
     "sname" : "rahul",
     "degree" : "BCA",
     "sem" : "2nd",
     "CGPA" : 7.9
}
```

5

```
    {
        "_id" : ObjectId("62930e3b4c0403136b43e946"),
        "SRN" : "R21DB044",
        "sname" : "rohith",
        "degree" : "BCA",
        "sem" : "5th",
        "CGPA" : 6.9
    }
```

xiii. Display all the students in BCA and in 6th Sem.

o/p

> db.stud.find({$and:[{degree:"BCA"},{sem:"5th"}]})

{ "_id" : ObjectId("62930e3b4c0403136b43e946"), "SRN" : "R21DB044", "sname" : "rohith", "degree" : "BCA", "sem" : "5th", "CGPA" : 6.9 }

2. Employee Database

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

i. Insert 10 documents.

o/p

{ "eid" : 38, "ename" : "rocky", "dept" : "production", "design" : "HR", "salary" : 40000, "yoj" : 2016, "address" : { "dno" : 48, "street" : 3, "locality" : "jayanagar", "cilt" : "bangalore" } }

{ "eid" : 46, "ename" : "jancy", "dept" : "developer", "design" : "TL", "salary" : 80000, "yoj" : 2014, "address" : { "dno" : 50, "street" : 3, "locality" : "indranagar", "cilt" : "bangalore" } }

{ "eid" : 99, "ename" : "rahul", "dept" : "developer", "design" : "TL", "salary" : 45000, "yoj" : 2014, "address" : { "dno" : 99, "street" : 3, "locality" : "kormangala", "cilt" : "bangalore" } }

{ "eid" : 54, "ename" : "raghu", "dept" : "developer", "design" : "manager", "salary" : 95000, "yoj" : 2012, "address" : { "dno" : 55, "street" : 3, "locality" : "dwarakanagar", "cilt" : "bangalore" } }

6

{ "eid" : 18, "ename" : "mohan", "dept" : "production", "design" : "fresher", "salary" : 35000, "yoj" : 2012, "address" : { "dno" : 108, "street" : 33, "locality" : "bagalore", "cilt" : "bangalore" } }

{ "eid" : 68, "ename" : "kiran", "dept" : "developer", "design" : "intern", "salary" : 35000, "yoj" : 2022, "address" : { "dno" : 44, "street" : 33, "locality" : "bagalore", "cilt" : "bangalore" } }

{ "eid" : 37, "ename" : "mohith", "dept" : "developer", "design" : "intern", "salary" : 35000, "yoj" : 2022, "address" : { "dno" : 44, "street" : 33, "locality" : "bagalore", "cilt" : "bangalore" } }

{ "eid" : 10, "ename" : "mithun", "dept" : "developer", "design" : "intern", "salary" : 35000, "yoj" : 2022, "address" : { "dno" : 77, "street" : 33, "locality" : "bagalore", "cilt" : "bangalore" } }

{ "eid" : 10, "ename" : "mithun", "dept" : "developer", "design" : "intern", "salary" : 35000, "yoj" : 2022, "address" : { "dno" : 77, "street" : 33, "locality" : "bagalore", "cilt" : "bangalore" } }

{ "eid" : 89, "ename" : "lambu", "dept" : "developer", "design" : "intern", "salary" : 35000, "yoj" : 2022, "address" : { "dno" : 77, "street" : 33, "locality" : "bagalore", "cilt" : "bangalore" } }

ii. Display all the employees with salary in range (50000, 75000).

o/p

> db.emp.find({$and:[{salary:{$gt:40000}},{salary:{$lt:75000}}]})

{ "_id" : ObjectId("629341a0df92f1e71373671f"), "eid" : 99, "ename" : "rahul", "dept" : "developer", "design" : "TL", "salary" : 45000, "yoj" : 2014, "address" : { "dno" : 99, "street" : 3, "locality" : "kormangala", "cilt" : "bangalore" } }

iii. Display all the employees with department developer".

o/p

> db.emp.find({dept:"developer"},{ename:1,dept:1,_id:0})

{ "ename" : "jancy", "dept" : "developer" }

{ "ename" : "rahul", "dept" : "developer" }

{ "ename" : "raghu", "dept" : "developer" }

{ "ename" : "kiran", "dept" : "developer" }

{ "ename" : "mohith", "dept" : "developer" }

{ "ename" : "mithun", "dept" : "developer" }

{ "ename" : "mithun", "dept" : "developer" }

{ "ename" : "lambu", "dept" : "developer" }

iv. Display the Salary of "Rahul".

o/p

> db.emp.find({ename:"rahul"},{ename:1,salary:1,_id:0})

{ "ename" : "rahul", "salary" : 45000 }

v. Display the city of employee "Rahul".

o/p

> db.emp.find({ename:"rahul"},{ename:1,address:{cilt:1},_id:0})

{ "ename" : "rahul", "address" : { "cilt" : "bangalore" } }

vi. Update the salary of developers by 5000 increment .

o/p

> db.emp.update({dept:"developer"},{$inc:{salary:5000}},{multi:true})

WriteResult({ "nMatched" : 8, "nUpserted" : 0, "nModified" : 8 })

vii. Add field age to employee "Rahul".

o/p

> db.emp.update({ename:"rahul"},{$set:{age:22}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

{
  "_id" : ObjectId("629341a0df92f1e71373671f"),
  "eid" : 99,
  "ename" : "rahul",
  "dept" : "developer",
  "design" : "TL",
  "salary" : 50000,
  "yoj" : 2014,
  "address" : {
    "dno" : 99,
    "street" : 3,
    "locality" : "kormangala",
    "cilt" : "bangalore"
  },
  "age" : 22

```
        }
```

viii. Remove YOJ from "Rahul".

o/p

```
> db.emp.update({ename:"rahul"},{$unset:{yoj:1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
{
        "_id" : ObjectId("629341a0df92f1e71373671f"),
        "eid" : 99,
        "ename" : "rahul",
        "dept" : "developer",
        "design" : "TL",
        "salary" : 50000,
        "address" : {
                "dno" : 99,
                "street" : 3,
                "locality" : "kormangala",
                "cilt" : "bangalore"
        },
        "age" : 22
}
```

ix. Add an array field project to "Rahul".

o/p

```
> db.emp.update({ename:"rahul"},{$push:{projects:"p1"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
{
        "_id" : ObjectId("629341a0df92f1e71373671f"),
        "eid" : 99,
        "ename" : "rahul",
        "dept" : "developer",
        "design" : "TL",
        "salary" : 50000,
        "address" : {
```

```
            "dno" : 99,
            "street" : 3,
            "locality" : "kormangala",
            "cilt" : "bangalore"
        },
        "age" : 22,
        "projects" : [
            "p1"
        ]
    }
```

x. Add p2 and p3 project to "Rahul".

```
o/p
> db.emp.update({ename:"rahul"},{$push:{projects:{$each:["p2","p3"]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
{
        "_id" : ObjectId("629341a0df92f1e71373671f"),
        "eid" : 99,
        "ename" : "rahul",
        "dept" : "developer",
        "design" : "TL",
        "salary" : 50000,
        "address" : {
            "dno" : 99,
            "street" : 3,
            "locality" : "kormangala",
            "cilt" : "bangalore"
        },
        "age" : 22,
        "projects" : [
            "p1",
            "p2",
            "p3"
        ]
```

```
        }

xi.  Remove p3 from "Rahul".
     o/p
     > db.emp.update({ename:"rahul"},{$pull:{projects:"p3"}})
     WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
     {
          "_id" : ObjectId("629341a0df92f1e71373671f"),
          "eid" : 99,
          "ename" : "rahul",
          "dept" : "developer",
          "design" : "TL",
          "salary" : 50000,
          "address" : {
               "dno" : 99,
               "street" : 3,
               "locality" : "kormangala",
               "cilt" : "bangalore"
          },
          "age" : 22,
          "projects" : [
               "p1",
               "p2"
          ]
     }

xii. Add a new embedded object "contacts" with "email" and "phone" as array objects to "Rahul".
     o/p
     > db.emp.update({ename:"rahul"},{$push:{contacts:{phone:9976553219,email:"rahul@gmail.com"}}})
     WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
     {
          "_id" : ObjectId("629341a0df92f1e71373671f"),
          "eid" : 99,
          "ename" : "rahul",
```

11

```
        "dept" : "developer",
        "design" : "TL",
        "salary" : 50000,
        "address" : {
            "dno" : 99,
            "street" : 3,
            "locality" : "kormangala",
            "cilt" : "bangalore"
        },
        "age" : 22,
        "projects" : [
            "p1",
            "p2"
        ],
        "contacts" : [
            {
                "phone" : 9976553219,
                "email" : "rahul@gmail.com"
            }
        ]
    }
```

xiii. Add two phone numbers to "Rahul".

```
o/p
> db.emp.update({ename:"rahul"},{$addToSet:{contacts:{phone:[9901180740,9901180440]}}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
{
    "_id" : ObjectId("629341a0df92f1e71373671f"),
    "eid" : 99,
    "ename" : "rahul",
    "dept" : "developer",
    "design" : "TL",
    "salary" : 50000,
    "address" : {
```

12

```
        "dno" : 99,
        "street" : 3,
        "locality" : "kormangala",
        "cilt" : "bangalore"
    },
    "age" : 22,
    "projects" : [
        "p1",
        "p2"
    ],
    "contacts" : [
        {
            "phone" : 9976553219,
            "email" : "rahul@gmail.com"
        },
        {
            "phone" : [
                9901180740,
                9901180440
            ]
        }
    ]
}
```

3. Book Database

Create a book Data Base with the fields: (isbn, bname, author [], year, publisher, price)

i. Insert 5 documents.

```
o/p
>
db.book.insert({isbn:345619,bname:"python",authore:["herbet","kuvempu"],year:2000,publisher:"pearson",price:499})
WriteResult({ "nInserted" : 1 })
```

```
>
db.book.insert({isbn:322619,bname:"mongodb",authore:["herbet","rudresh"],year:2001,publisher:"lalitha",pr
ice:789})
WriteResult({ "nInserted" : 1 })
> db.book.insert({isbn:999619,bname:"java",authore:["rama"],year:2021,publisher:"pearson",price:229})
WriteResult({ "nInserted" : 1 })
>db.book.insert({isbn:12389,bname:"let_us_see",authore:["rudresh","kuvempu"],year:1998,publisher:"pears
on",price:599})
WriteResult({ "nInserted" : 1 })
> doc2 = {isbn:12453,bname:"java",authore:["rama","mohith"],year:2016,publisher:"nandhini",price:368}
> db.book.insert("doc2")
```

ii. List all the documents.

```
o/p
> db.book.find().pretty()
{
     "_id" : ObjectId("6294e65ed3a961fc352bf202"),
     "isbn" : 12453,
     "bname" : "java",
     "authore" : [
          "rama",
          "mohith"
     ],
     "year" : 2016,
     "publisher" : "nandhini",
     "price" : 368
}
{
     "_id" : ObjectId("6294e7cdd3a961fc352bf203"),
     "isbn" : 12389,
     "bname" : "let us see",
     "authore" : [
          "rudresh",
          "kuvempu"
```

14

```
        ],
        "year" : 1998,
        "publisher" : "pearson",
        "price" : 599
    }
    {
        "_id" : ObjectId("6294e82ed3a961fc352bf204"),
        "isbn" : 345619,
        "bname" : "python",
        "authore" : [
            "herbet",
            "kuvempu"
        ],
        "year" : 2000,
        "publisher" : "pearson",
        "price" : 499
    }
    {
        "_id" : ObjectId("6294e87bd3a961fc352bf205"),
        "isbn" : 322619,
        "bname" : "mongodb",
        "authore" : [
            "herbet",
            "rudresh"
        ],
        "year" : 2001,
        "publisher" : "lalitha",
        "price" : 789
    }
    {
        "_id" : ObjectId("6294e921d3a961fc352bf206"),
        "isbn" : 999619,
        "bname" : "java",
        "authore" : [
```

```
                "rama"
            ],
            "year" : 2021,
            "publisher" : "pearson",
            "price" : 229
    }
```

iii. List all book names except year and price.

o/p

> db.book.find({},{year:0,price:0})

{ "_id" : ObjectId("6294e65ed3a961fc352bf202"), "isbn" : 12453, "bname" : "java", "authore" : ["rama", "mohith"], "publisher" : "nandhini" }

{ "_id" : ObjectId("6294e7cdd3a961fc352bf203"), "isbn" : 12389, "bname" : "let us see", "authore" : ["rudresh", "kuvempu"], "publisher" : "pearson" }

{ "_id" : ObjectId("6294e82ed3a961fc352bf204"), "isbn" : 345619, "bname" : "python", "authore" : ["herbet", "kuvempu"], "publisher" : "pearson" }

{ "_id" : ObjectId("6294e87bd3a961fc352bf205"), "isbn" : 322619, "bname" : "mongodb", "authore" : ["herbet", "rudresh"], "publisher" : "lalitha" }

{ "_id" : ObjectId("6294e921d3a961fc352bf206"), "isbn" : 999619, "bname" : "java", "authore" : ["rama"], "publisher" : "pearson" }

iv. Display all the books authored by rudresh.

o/p

> db.book.find({authore:"rudresh"})

{ "_id" : ObjectId("6294e7cdd3a961fc352bf203"), "isbn" : 12389, "bname" : "let us see", "authore" : ["rudresh", "kuvempu"], "year" : 1998, "publisher" : "pearson", "price" : 599 }

{ "_id" : ObjectId("6294e87bd3a961fc352bf205"), "isbn" : 322619, "bname" : "mongodb", "authore" : ["herbet", "rudresh"], "year" : 2001, "publisher" : "lalitha", "price" : 789 }

v. List all the books published by pearson.

o/p

> db.book.find({publisher:"pearson"})

{ "_id" : ObjectId("6294e7cdd3a961fc352bf203"), "isbn" : 12389, "bname" : "let us see", "authore" : ["rudresh", "kuvempu"], "year" : 1998, "publisher" : "pearson", "price" : 599 }

{ "_id" : ObjectId("6294e82ed3a961fc352bf204"), "isbn" : 345619, "bname" : "python", "authore" : ["herbet", "kuvempu"], "year" : 2000, "publisher" : "pearson", "price" : 499 }

{ "_id" : ObjectId("6294e921d3a961fc352bf206"), "isbn" : 999619, "bname" : "java", "authore" : ["rama"], "year" : 2021, "publisher" : "pearson", "price" : 229 }

vi. List the publisher of book java.

o/p

> db.book.find({bname:"java"},{publisher:1,bname:1,_id:0})

{ "bname" : "java", "publisher" : "nandhini" }

{ "bname" : "java", "publisher" : "pearson" }

vii. List the author, publisher and year of the book let us see.

o/p

> db.book.find({bname:"let us see"},{publisher:1,bname:1,authore:1,year:1,_id:0})

{ "bname" : "let us see", "authore" : ["rudresh", "kuvempu"], "year" : 1998, "publisher" : "pearson" }

viii. Display the price of "let us see" except _id.

o/p

> db.book.find({bname:"let us see"},{price:1,bname:1,_id:0})

{ "bname" : "let us see", "price" : 599 }

ix. Sort and display all books in ascending order of book names.

o/p

> db.book.find({},{bname:1,_id:0}).sort({bname:1})

{ "bname" : "java" }

{ "bname" : "java" }

{ "bname" : "let us see" }

{ "bname" : "mongodb" }

{ "bname" : "python" }

x. Sort and display only 3 books in descending order of price.

o/p

> db.book.find({},{_id:0}).sort({price:-1}).limit(3)

17

{ "isbn" : 322619, "bname" : "mongodb", "authore" : ["herbet", "rudresh"], "year" : 2001, "publisher" : "lalitha", "price" : 789 }

{ "isbn" : 12389, "bname" : "let us see", "authore" : ["rudresh", "kuvempu"], "year" : 1998, "publisher" : "pearson", "price" : 599 }

{ "isbn" : 345619, "bname" : "python", "authore" : ["herbet", "kuvempu"], "year" : 2000, "publisher" : "pearson", "price" : 499 }

xi. Display all the books written by herbet and kuvempu.

o/p

xii. Display all the books either written by herbet and kuvempu.

xiii. Display all the books where rama is the first author.

4. Food Database

Create a Food Database with the fields: (food id, food cat, food name, chef name [ ], price, ingredients [], hotel name, hotel address {no, street, locality, city})

i. Insert 10 documents.

ii. List the price of pizza with ingredients.

iii. Display the item in the price range(500,800).

iv. Display the item prepared by x and y.

v. Display the item prepared by x or y.

vi. Add one chef to the food pizza.

vii. Add ingredients to the food Burger.

viii. Delete last ingredient added to the food burger.

ix. Delete all the ingredients from the food biryani.

x. Add food type to the food Burger.

xi. Modify the burger price by 200.

xii. Add or insert a new food item with the food Id "f08 " usingupsert as True.

18

xiii. Increment the price of all food item in food cat: fastfood by 120.

5. Import and export Bigdata to MongoDB

PART B

PHP with MONGODB

1. Demonstrate how to establish connection between PHP and MongoDB.

2. Grouping Data with Map/Reduce

3. Create Employee Database (PHP) and perform following operations.

   i. Connect to MongoDB.

   ii. Insert 5 documents into the employee database.

   iii. Find all documents in the database.

   iv. Find one document with condition.

   v. Display two Documents in the database using LIMIT Command.

   vi. Display from 5th document.

   vii. Sort the documents in Ascending order based on pin.

   viii. Display the prescribed number in an array object using SLICE operator.

   ix. Display the prescribed number in an array object using SLICE with SKIP-LIMIT.

4. Create Employee Database (PHP) and perform following operations.

   i. Connect to MongoDB.

   ii. Insert 5 documents into the employee database.

   iii. Display find with condition (where)

   iv. Demonstrate OR condition, AND condition, Conditional operators lt,lte,gt,gte,ne,
       in operator, all operator, EXISTS operator-checks whether field has a value.

5. Demonstrate Indexing in MongoDB.