LAB ASSIGNMENT – 2

Name: DishaBiswas

Roll No: 004 (DA)

DATE:19/01/2024

Q1. Make a schema first and then insert 6 documents.

- Roll_no=[1,2,3,4,5,]
- Name=["Ram","Alex","John","Bob","Mukesh","Danny"]
- Age=[20,19,40,55,30,28]
- Salary=[546.7,333.4,666.7,678.4,245.6,546.3]
- City=["A","B","C","D","E","F"]
- Phone_no=[123,456,122,444,567,892]

Query: db.students.insertMany([

... {roll_no: 1, name: "Ram", age: 20, salary: 546.7, city: "A", phone_no: 123},

... {roll_no: 2, name: "Alex", age: 19, salary: 333.4, city: "B", phone_no: 456},

... {roll_no: 3, name: "John", age: 40, salary: 666.7, city: "C", phone_no: 122},

... {roll_no: 4, name: "Bob", age: 55, salary: 678.4, city: "D", phone_no: 444},

... {roll_no: 5, name: "Mukesh", age: 30, salary: 245.6, city: "E", phone_no: 567},

... {roll_no: 6, name: "Danny", age: 28, salary: 546.3, city: "F", phone_no: 892}

...])

Output: {

acknowledged: true,

insertedIds: {

'0': ObjectId('697611febf3c95f1a01e2621'),

'1': ObjectId('697611febf3c95f1a01e2622'),

'2': ObjectId('697611febf3c95f1a01e2623'),

'3': ObjectId('697611febf3c95f1a01e2624'),

'4': ObjectId('697611febf3c95f1a01e2625'),

'5': ObjectId('697611febf3c95f1a01e2626')

}

}

Q2. Write a query to update the name of RAM to SAM. Query:

```
db.students.updateOne(
... {name: "Ram"},
... {$set:{name: "Sam"}}
...)
```

Output: {

```
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Q3. Write a query to display only the cities present in that collection. Query:

```
db.students.find({},{city: 1, _id: 0})
```

Output:[{ city: 'A' },

```
  {city: 'B'},
  {city: 'C'},
  {city: 'D'},
  { city: 'E' },
  { city: 'F' }]
```

Q4. Write a query to update the salary by 10%.

Query: db.students.updateMany({},{$mul:{salary: 1.1}}) Output: {

```
  acknowledged: true,
  insertedId: null,
  matchedCount: 6,
  modifiedCount: 6,
  upsertedCount: 0
}
```

Q5. Write a query to display all the documents in ascending and descending order of age. Query:

db.students.find().sort({age: 1})

Output: [{

_id: ObjectId('697611febf3c95f1a01e2622'),

roll_no: 2,

name: 'Alex', age:

19,

salary: 366.74,

city: 'B',

phone_no: 456

},

{

_id: ObjectId('697611febf3c95f1a01e2621'),

roll_no: 1,

name: 'Sam',

age: 20,

salary: 601.3700000000001,

city: 'A',

phone_no: 123

},

{

_id: ObjectId('697611febf3c95f1a01e2626'),

roll_no: 6,

name: 'Danny',

age: 28,

salary: 600.93,

city: 'F',

phone_no: 892

},

{

_id: ObjectId('697611febf3c95f1a01e2625'),

roll_no: 5, name:
'Mukesh', age: 30,
salary: 270.16,
city: 'E',
phone_no: 567
},
{
_id: ObjectId('697611febf3c95f1a01e2623'),
roll_no: 3,
name: 'John',
age: 40,
salary: 733.3700000000001,
city: 'C',
phone_no: 122
},
{
_id: ObjectId('697611febf3c95f1a01e2624'),
roll_no: 4,
name: 'Bob', age:
55,
salary: 746.24,
city: 'D',
phone_no: 444
}]

Query: db.students.find().sort({age: -1}) Output:

[
{
_id: ObjectId('697611febf3c95f1a01e2624'),
roll_no: 4,

```
  name: 'Bob', age:
  55,
  salary: 746.24,
  city: 'D',
  phone_no: 444
},
{
  _id: ObjectId('697611febf3c95f1a01e2623'),
  roll_no: 3,
  name: 'John',
  age: 40,
  salary: 733.3700000000001,
  city: 'C',
  phone_no: 122
},
{
  _id: ObjectId('697611febf3c95f1a01e2625'),
  roll_no: 5,
  name: 'Mukesh', age:
  30,
  salary: 270.16,
  city: 'E',
  phone_no: 567
},
{
  _id: ObjectId('697611febf3c95f1a01e2626'),
  roll_no: 6,
  name: 'Danny',
  age: 28,
  salary: 600.93,
  city: 'F',
```

phone_no: 892

},

{

_id: ObjectId('697611febf3c95f1a01e2621'),

roll_no: 1,

name: 'Sam',

age: 20,

salary: 601.3700000000001,

city: 'A',

phone_no: 123

},

{

_id: ObjectId('697611febf3c95f1a01e2622'),

roll_no: 2,

name: 'Alex', age:

19,

salary: 366.74,

city: 'B',

phone_no: 456

}]

Q6. Write a query to display all the documents with City → A,B,C. Query:

db.students.find({ city: { $in: ["A", "B", "C"] } })

Output: [{

_id: ObjectId('697611febf3c95f1a01e2621'),

roll_no: 1,

name: 'Sam',

age: 20,

salary: 601.3700000000001,

city: 'A',

phone_no: 123

},

{

_id: ObjectId('697611febf3c95f1a01e2622'),

roll_no: 2,

name: 'Alex', age:

19,

salary: 366.74,

city: 'B',

phone_no: 456

},

{

_id: ObjectId('697611febf3c95f1a01e2623'),

roll_no: 3,

name: 'John',

age: 40,

salary: 733.3700000000001,

city: 'C',

phone_no: 122

}

]

Q7. Write a query to display only two documents from the entire collection. Query:

db.students.find().limit(2)

Output: [{

_id: ObjectId('697611febf3c95f1a01e2621'),

roll_no: 1,

name: 'Sam',

age: 20,

salary: 601.3700000000001,

city: 'A',

phone_no: 123

},

{

_id: ObjectId('697611febf3c95f1a01e2622'),

roll_no: 2,

name: 'Alex', age:

19,

salary: 366.74,

city: 'B',

phone_no: 456

}]

Q8. Write a query to delete a document with ROLL_NO:5. Query:

db.students.deleteOne({ roll_no: 5 })

Output: { acknowledged: true, deletedCount: 1 }

Q9. Write a query to display all the documents with AGE greater than 20. Query:

db.students.find({ age: { $gt: 20 } })

Output: [{

_id: ObjectId('697611febf3c95f1a01e2623'),

roll_no: 3,

name: 'John',

age: 40,

salary: 733.3700000000001,

city: 'C',

phone_no: 122

},

{

_id: ObjectId('697611febf3c95f1a01e2624'),

roll_no: 4,

name: 'Bob', age:

55,

salary: 746.24,

city: 'D',

phone_no: 444

},

{

_id: ObjectId('697611febf3c95f1a01e2626'),

roll_no: 6,

name: 'Danny',

age: 28,

salary: 600.93,

city: 'F',

phone_no: 892

}]

Q10. Write a query to display all the documents with AGE less than 20. Query:

db.students.find({age: {$lt: 20}})

Output: [{

_id: ObjectId('697611febf3c95f1a01e2622'),

roll_no: 2,

name: 'Alex', age:

19,

salary: 366.74,

city: 'B',

phone_no: 456

}]

Q11. Write a query to display all the documents with AGE equals to 20. Query:

db.students.find({age: 20})

Output: [{

_id: ObjectId('697611febf3c95f1a01e2621'),

roll_no: 1,

name: 'Sam',

age: 20,

salary: 601.3700000000001,

city: 'A',

phone_no: 123

}]

Q12. Write a query to display all the documents with AGE not equals to 20. Query:

db.students.find({age: {$ne: 20}})

Output: [

{

_id: ObjectId('697611febf3c95f1a01e2622'),

roll_no: 2,

name: 'Alex', age:

19,

salary: 366.74,

city: 'B',

phone_no: 456

},

{

_id: ObjectId('697611febf3c95f1a01e2623'),

roll_no: 3,

name: 'John',

age: 40,

salary: 733.3700000000001,

city: 'C',

phone_no: 122

},

{

_id: ObjectId('697611febf3c95f1a01e2624'),

roll_no: 4,

name: 'Bob', age:

55,

salary: 746.24,

city: 'D',

phone_no: 444

},

{

```
  _id: ObjectId('697611febf3c95f1a01e2626'),
  roll_no: 6,
  name: 'Danny',
  age: 28,
  salary: 600.93,
  city: 'F',
  phone_no: 892
 }
]
```

Q13. Write a query to display all the documents where AGE is greater than equals to 30. Query:

```
db.students.find({ age: { $gte: 30 } })
```

Output:

```
[
 {
  _id: ObjectId('697611febf3c95f1a01e2623'),
  roll_no: 3,
  name: 'John',
  age: 40,
  salary: 733.3700000000001,
  city: 'C',
  phone_no: 122
 },
 {
  _id: ObjectId('697611febf3c95f1a01e2624'),
  roll_no: 4,
  name: 'Bob', age:
  55,
  salary: 746.24,
  city: 'D',
  phone_no: 444
 }
]
```

```
_id: ObjectId('697611febf3c95f1a01e2621')
roll_no : 1
name : "Sam"
age : 20
salary : 601.3700000000001
city : "A"
phone_no : 123
```

```
_id: ObjectId('697611febf3c95f1a01e2622')
roll_no : 2
name : "Alex"
age : 19
salary : 366.74
city : "B"
phone_no : 456
```

```
_id: ObjectId('697611febf3c95f1a01e2623')
roll_no : 3
name : "John"
age : 40
salary : 733.3700000000001
city : "C"
phone_no : 122
```

```
_id: ObjectId('697611febf3c95f1a01e2624')
roll_no : 4
name : "Bob"
age : 55
salary : 746.24
city : "D"
phone_no : 444
```

```
_id: ObjectId('697611febf3c95f1a01e2626')
roll_no : 6
name : "Danny"
age : 28
salary : 600.93
city : "F"
phone_no : 892
```