MONGO DB PROJECT 1

Q.1)

```
db.test.insert({
   "_id" : 1,
   "name" : {
      "first" : "John",
      "last" : "Backus"
   },
   "birth" : ISODate("1924-12-03T05:00:00Z"),
   "death" : ISODate("2007-03-17T04:00:00Z"),
   "contribs" : [
      "Fortran",
      "ALGOL",
      "Backus-Naur Form",
      "FP"
   ],
   "awards" : [
      {
         "award" : "W.W. McDowell Award",
         "year" : 1967,
         "by" : "IEEE Computer Society"
      },
      {
         "award" : "National Medal of Science",
         "year" : 1975,
         "by" : "National Science Foundation"
      },
```

```
      {
         "award" : "Turing Award",
         "year" : 1977,
         "by" : "ACM"
      },
      {
         "award" : "Draper Prize",
         "year" : 1993,
         "by" : "National Academy of Engineering"
      }
   ]
})

db.test.insert({
   "_id" : ObjectId("51df07b094c6acd67e492f41"),
   "name" : {
      "first" : "John",
      "last" : "McCarthy"
   },
   "birth" : ISODate("1927-09-04T04:00:00Z"),
   "death" : ISODate("2011-12-24T05:00:00Z"),
   "contribs" : [
      "Lisp",
      "Artificial Intelligence",
      "ALGOL"
   ],
   "awards" : [
      {
```

```
            "award" : "Turing Award",
            "year" : 1971,
            "by" : "ACM"
        },
        {
            "award" : "Kyoto Prize",
            "year" : 1988,
            "by" : "Inamori Foundation"
        },
        {
            "award" : "National Medal of Science",
            "year" : 1990,
            "by" : "National Science Foundation"
        }
    ]
})

db.test.insert({
    "_id" : 3,
    "name" : {
        "first" : "Grace",
        "last" : "Hopper"
    },
    "title" : "Rear Admiral",
    "birth" : ISODate("1906-12-09T05:00:00Z"),
    "death" : ISODate("1992-01-01T05:00:00Z"),
    "contribs" : [
        "UNIVAC",
```

```
    "compiler",
    "FLOW-MATIC",
    "COBOL"
  ],
  "awards" : [
    {
      "award" : "Computer Sciences Man of the Year",
      "year" : 1969,
      "by" : "Data Processing Management Association"
    },
    {
      "award" : "Distinguished Fellow",
      "year" : 1973,
      "by" : " British Computer Society"
    },
    {
      "award" : "W. W. McDowell Award",
      "year" : 1976,
      "by" : "IEEE Computer Society"
    },
    {
      "award" : "National Medal of Technology",
      "year" : 1991,
      "by" : "United States"
    }
  ]
})
```

```
db.test.insert({
   "_id" : 4,
   "name" : {
      "first" : "Kristen",
      "last" : "Nygaard"
   },
   "birth" : ISODate("1926-08-27T04:00:00Z"),
   "death" : ISODate("2002-08-10T04:00:00Z"),
   "contribs" : [
      "OOP",
      "Simula"
   ],
   "awards" : [
      {
         "award" : "Rosing Prize",
         "year" : 1999,
         "by" : "Norwegian Data Association"
      },
      {
         "award" : "Turing Award",
         "year" : 2001,
         "by" : "ACM"
      },
      {
         "award" : "IEEE John von Neumann Medal",
         "year" : 2001,
         "by" : "IEEE"
      }
```

```
    ]
})

db.test.insert({
    "_id" : 5,
    "name" : {
        "first" : "Ole-Johan",
        "last" : "Dahl"
    },
    "birth" : ISODate("1931-10-12T04:00:00Z"),
    "death" : ISODate("2002-06-29T04:00:00Z"),
    "contribs" : [
        "OOP",
        "Simula"
    ],
    "awards" : [
        {
            "award" : "Rosing Prize",
            "year" : 1999,
            "by" : "Norwegian Data Association"
        },
        {
            "award" : "Turing Award",
            "year" : 2001,
            "by" : "ACM"
        },
        {
            "award" : "IEEE John von Neumann Medal",
```

```
            "year" : 2001,
            "by" : "IEEE"
        }
    ]
})

db.test.insert({
    "_id" : 6,
    "name" : {
        "first" : "Guido",
        "last" : "van Rossum"
    },
    "birth" : ISODate("1956-01-31T05:00:00Z"),
    "contribs" : [
        "Python"
    ],
    "awards" : [
        {
            "award" : "Award for the Advancement of Free Software",
            "year" : 2001,
            "by" : "Free Software Foundation"
        },
        {
            "award" : "NLUUG Award",
            "year" : 2003,
            "by" : "NLUUG"
        }
    ]
```

```
})

db.test.insert({
   "_id" : ObjectId("51e062189c6ae665454e301d"),
   "name" : {
      "first" : "Dennis",
      "last" : "Ritchie"
   },
   "birth" : ISODate("1941-09-09T04:00:00Z"),
   "death" : ISODate("2011-10-12T04:00:00Z"),
   "contribs" : [
      "UNIX",
      "C"
   ],
   "awards" : [
      {
         "award" : "Turing Award",
         "year" : 1983,
         "by" : "ACM"
      },
      {
         "award" : "National Medal of Technology",
         "year" : 1998,
         "by" : "United States"
      },
      {
         "award" : "Japan Prize",
         "year" : 2011,
```

```
            "by" : "The Japan Prize Foundation"
        }
    ]
})

db.test.insert({
    "_id" : 8,
    "name" : {
        "first" : "Yukihiro",
        "aka" : "Matz",
        "last" : "Matsumoto"
    },
    "birth" : ISODate("1965-04-14T04:00:00Z"),
    "contribs" : [
        "Ruby"
    ],
    "awards" : [
        {
            "award" : "Award for the Advancement of Free Software",
            "year" : "2011",
            "by" : "Free Software Foundation"
        }
    ]
})

db.test.insert({
    "_id" : 9,
    "name" : {
```

```
        "first" : "James",
        "last" : "Gosling"
    },
    "birth" : ISODate("1955-05-19T04:00:00Z"),
    "contribs" : [
        "Java"
    ],
    "awards" : [
        {
            "award" : "The Economist Innovation Award",
            "year" : 2002,
            "by" : "The Economist"
        },
        {
            "award" : "Officer of the Order of Canada",
            "year" : 2007,
            "by" : "Canada"
        }
    ]
})

db.test.insert({
    "_id" : 10,
    "name" : {
        "first" : "Martin",
        "last" : "Odersky"
    },
    "contribs" : [
```

```
      "Scala"
   ]
})
```

Q.2)

The normal update query did not work because it says “you cannot update the immutable object (_id)”

```
Doc = db.test.findOne({"name.first": "John", "name.last":"McCarthy"})
Doc._id = 2
db.test.insert(Doc)
db.test.remove({_id: ObjectId("51df07b094c6acd67e492f41")})
```

Q.3)

```
db.test.insert({
 "_id" : 20,
 "name" : {
 "first" : "Alex",
 "last" : "Chen"
 },
 "birth" : ISODate("1933-08-27T04:00:00Z"),
 "death" : ISODate("1984-11-07T04:00:00Z"),
 "contribs" : [
 "C++",
 "Simula"
 ],
 "awards" : [
 {
 "award" : "WPI Award",
```

```
 "year" : 1977,
 "by" : "WPI"
 }
 ]
})
db.test.insert({
 "_id" : 30,
 "name" : {
 "first" : "David",
 "last" : "Mark"
 },
 "birth" : ISODate("1911-04-12T04:00:00Z"),
 "death" : ISODate("2000-11-07T04:00:00Z"),
 "contribs" : [
 "C++",
 "FP",
 "Lisp",
 ],
 "awards" : [
 {
 "award" : "WPI Award",
 "year" : 1963,
 "by" : "WPI"
 },
 {
 "award" : "Turing Award",
 "year" : 1966,
 "by" : "ACM"
```

}

]

})

Q.4)

db.test.find({awards: {"$elemMatch": {award: "Turing Award", year: {$gt: 1976}

}}})

Q.5)

db.test.find({$or: [{awards: {$exists: true}, $where: "this.awards.length < 3"

}, {awards: {$exists: false}}, {contribs: "FP"}]})

(or)

db.test.find({$or:[{awards: {$size:0}},{awards: {$size:1}},{awards: {$size:2}},{contribs:"FP"}]})

Q.6)

db.test.find({"name.first": "Dennis", "name.last": "Ritchie"}, {name: 1, contr

ibs: 1, _id: 0})

Q.7)

db.test.update({"name.first": "Guido", "name.last": "van Rossum"}, {$push: {co

ntribs: "OOP"}})

Q.8)

db.test.update({"name.first": "Alex", "name.last": "Chen"},{$set: {"comments":

["He taught in 3 universities", "died from cancer", "lived in CA"]}})

(Or)

db.test.update({name: {first: "Alex", last:"Chen"}},{$push: {"comments": {$each:["He taught 3 uinversities","died from cancer","lived in CA"]}}})

Q.9)

```
db.test.aggregate(
{$unwind:"$contribs"},{$group:{_id:"$contribs",people:{$addToSet:"$name"}}},{$match:{people:{first : "Alex",
    last : "Chen"}}},{$sort:{_id:1,people:1}}
)
```

Q.10)

```
db.test.find({"name.first": {$regex: "Jo*"}}).sort({"name.last": 1})
```

Q.11)

```
db.test.distinct("awards.by")
```

Q.12)

```
db.test.update({}, {$unset: {death: ""}}, {multi: true})
```

Q.13)

```
db.test.update({ }, { $pull: {awards: {year: 2011}}}, {multi: true})
```

(or)

```
db.test.remove({"awards.year":2011})
```

Q.14)

```
db.test.update({_id: 30, "awards.by": "WPI"}, {$set: {"awards.$.year": 1965}})
```

Q.15)

```
db.test.find({_id: 3}).forEach( function(elem) { db.bios.update({_id: 30}, {$p
ushAll: {contribs: elem.contribs}})})
```

Q.16)

```
db.test.aggregate(
{$unwind:"$awards"},
{$match:{"awards.year":2001}},
{$project:{"name":1}},
{$group:{_id:"$name","count":{$sum:1}}},
{$match:{count:{$gt:1}}},
{$project:{_id:1}}
)
```

Q.17)

```
db.test.find().sort({_id: -1}).limit(1).pretty()
```

Q.18)

```
db.test.findOne({"awards.by": "ACM"})
```

Q.19)

```
db.test.remove({_id: {$in:[30,20]}})
```

Q.20)

```
db.test.count()
```

(or)

```
Db.test.find().count()
```
