Big Data and Analytical Lab

Lab Assignment – 08

(BCSE0183)

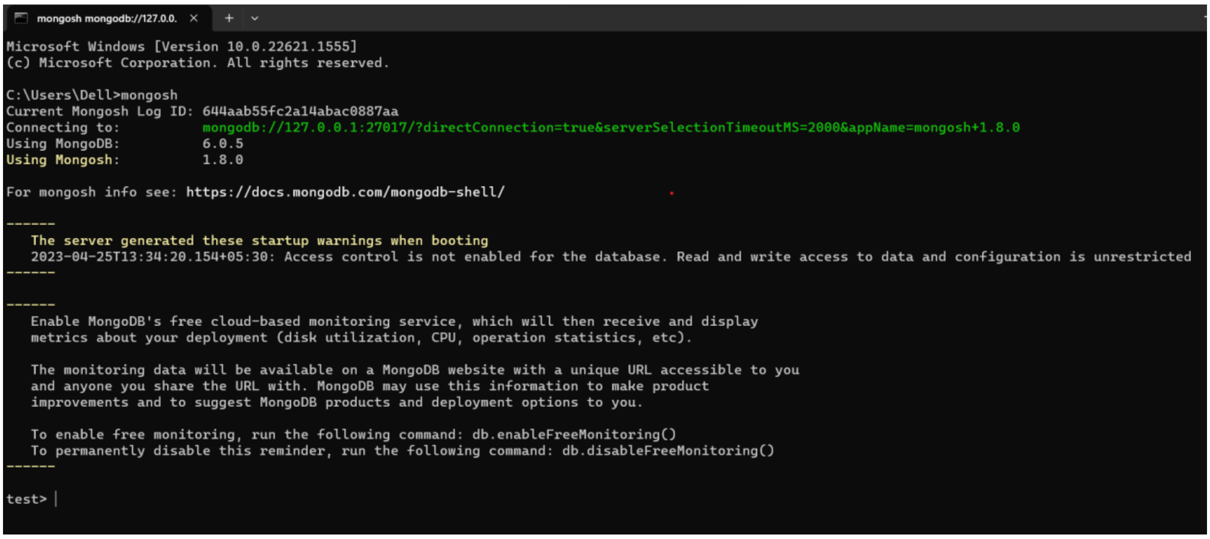
Made by:- Vishal Dixit

Sec- B (62)

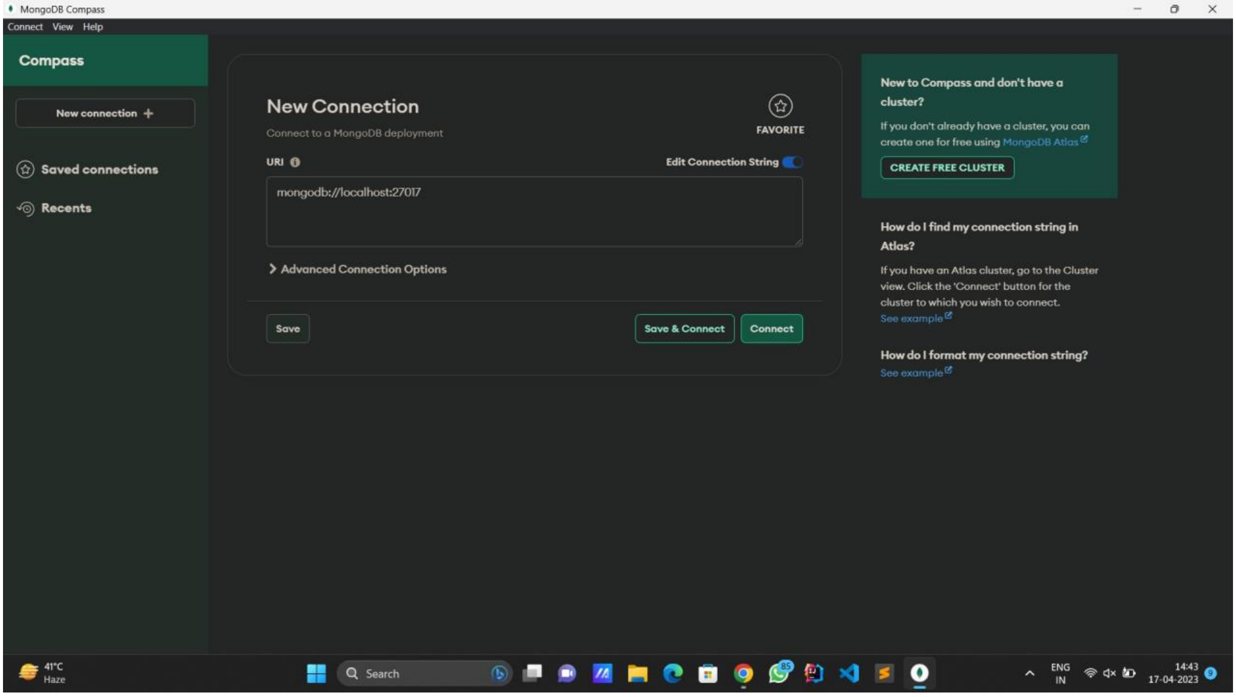
University Roll No :- 201500792

Mongo DB Commands

1) Installation of MongoDB and MongoDB Compass at your respectivemachine/system.



```
Microsoft Windows [Version 10.0.22621.1555]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Dell>mongosh
Current Mongosh Log ID: 644aab55fc2a14abac0887aa
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.8.0
Using MongoDB:          6.0.5
Using Mongosh:          1.8.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-04-25T13:34:20.154+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------

test>
```



New Connection

Connect to a MongoDB deployment

URI

Edit Connection String

mongodb://localhost:27017

Advanced Connection Options

New to Compass and don't have a cluster?

If you don't already have a cluster, you can create one for free using MongoDB Atlas

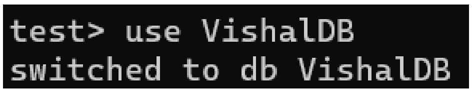
How do I find my connection string in Atlas?

If you have an Atlas cluster, go to the Cluster view. Click the 'Connect' button for the cluster to which you wish to connect.
See example

How do I format my connection string?
See example

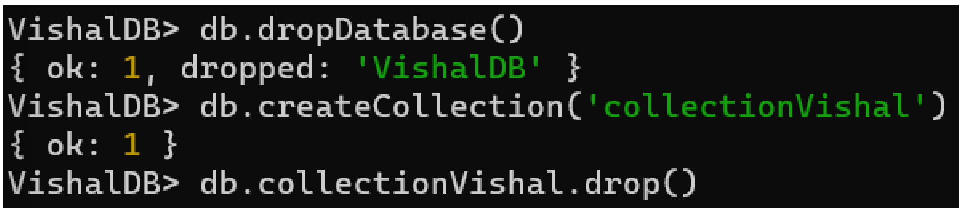
2) Implement the following MongoDB commands:

a. createDatabase ()



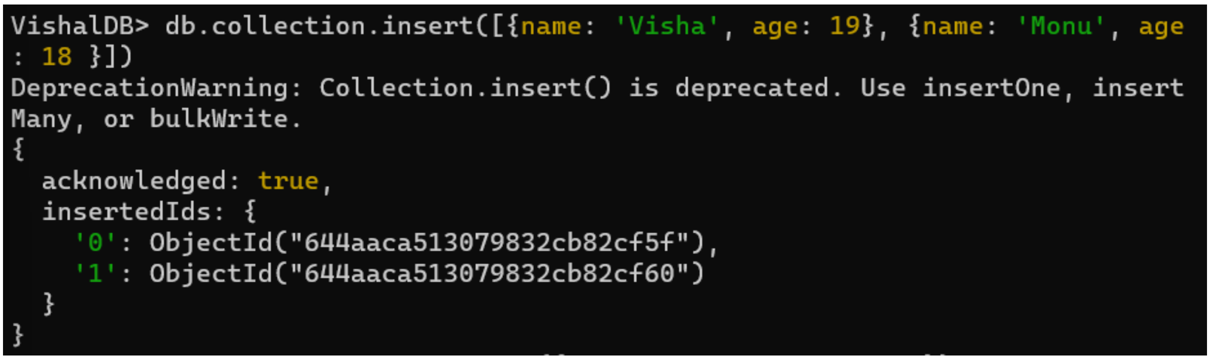
```
test> use VishalDB
switched to db VishalDB
```

b. dropDatabase ()



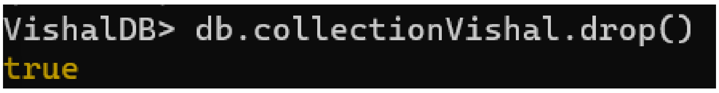
```
VishalDB> db.dropDatabase()
{ ok: 1, dropped: 'VishalDB' }
VishalDB> db.createCollection('collectionVishal')
{ ok: 1 }
VishalDB> db.collectionVishal.drop()
```

c. createCollection ()



```
VishalDB> db.collection.insert([{name: 'Visha', age: 19}, {name: 'Monu', age
: 18 }])
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insert
Many, or bulkWrite.
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("644aaca513079832cb82cf5f"),
    '1': ObjectId("644aaca513079832cb82cf60")
  }
}
```

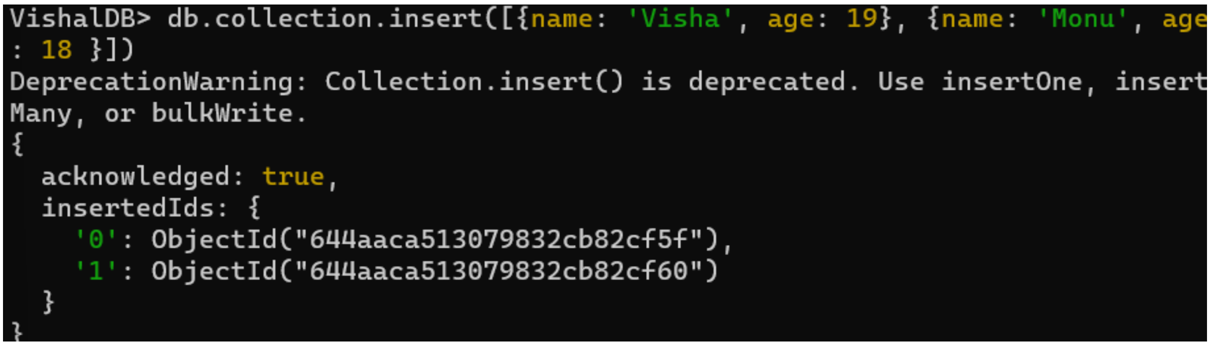
d. drop ()



```
VishalDB> db.collectionVishal.drop()
true
```

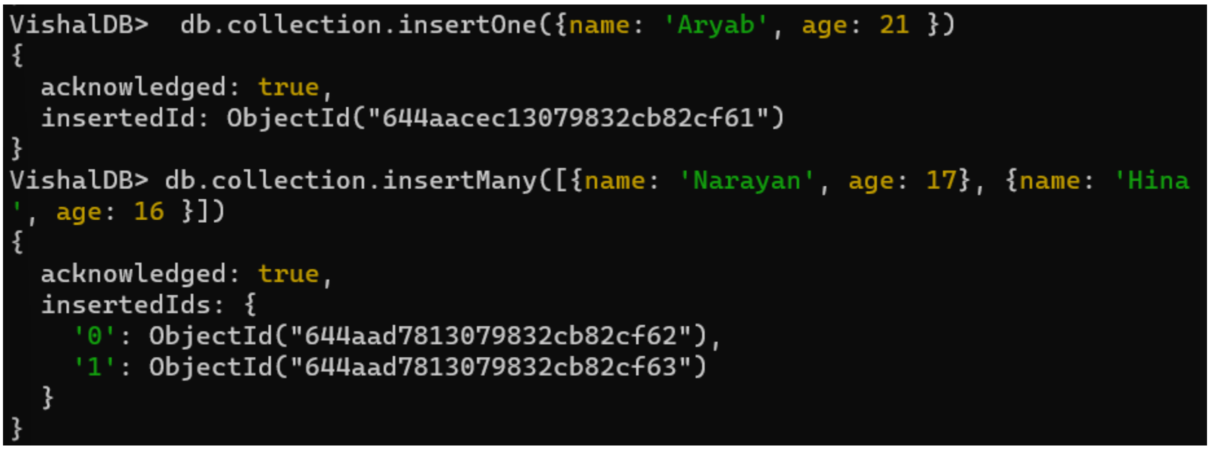
3) Insertion Operations –

a. Insert()



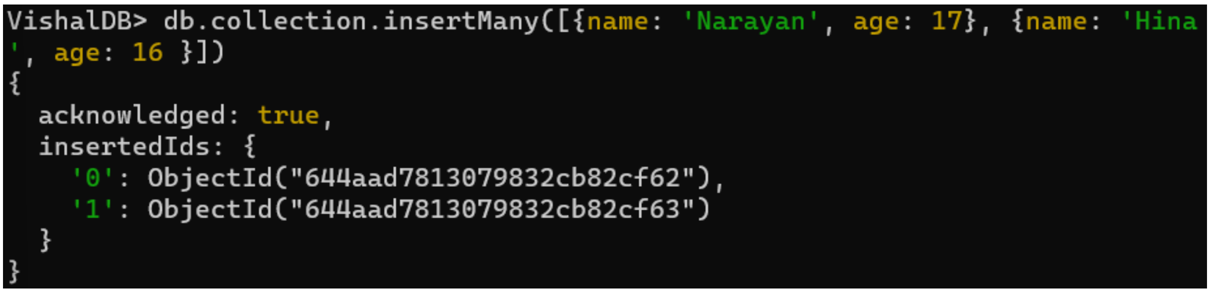
```
VishalDB> db.collection.insert([{name: 'Visha', age: 19}, {name: 'Monu', age
: 18 }])
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insert
Many, or bulkWrite.
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("644aaca513079832cb82cf5f"),
    '1': ObjectId("644aaca513079832cb82cf60")
  }
}
```

b. **insertOne()**



```
VishalDB>  db.collection.insertOne({name: 'Aryab', age: 21 })
{
  acknowledged: true,
  insertedId: ObjectId("644aacec13079832cb82cf61")
}
VishalDB> db.collection.insertMany([{name: 'Narayan', age: 17}, {name: 'Hina
', age: 16 }])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("644aad7813079832cb82cf62"),
    '1': ObjectId("644aad7813079832cb82cf63")
  }
}
```

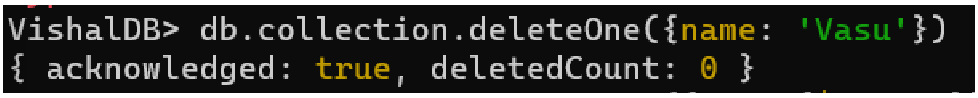
c. **insertMany()**



```
VishalDB> db.collection.insertMany([{name: 'Narayan', age: 17}, {name: 'Hina
', age: 16 }])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("644aad7813079832cb82cf62"),
    '1': ObjectId("644aad7813079832cb82cf63")
  }
}
```

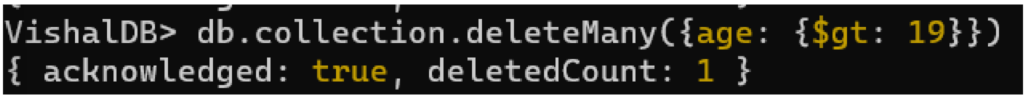
4) Deletion Operations-

a. **Delete ()**
b. **deleteOne ()**



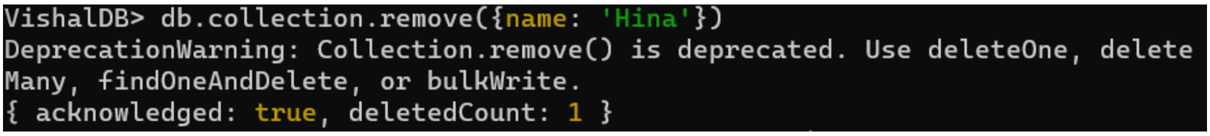
```
VishalDB> db.collection.deleteOne({name: 'Vasu'})
{ acknowledged: true, deletedCount: 0 }
```

c. **deleteMany ()**



```
VishalDB> db.collection.deleteMany({age: {$gt: 19}})
{ acknowledged: true, deletedCount: 1 }
```

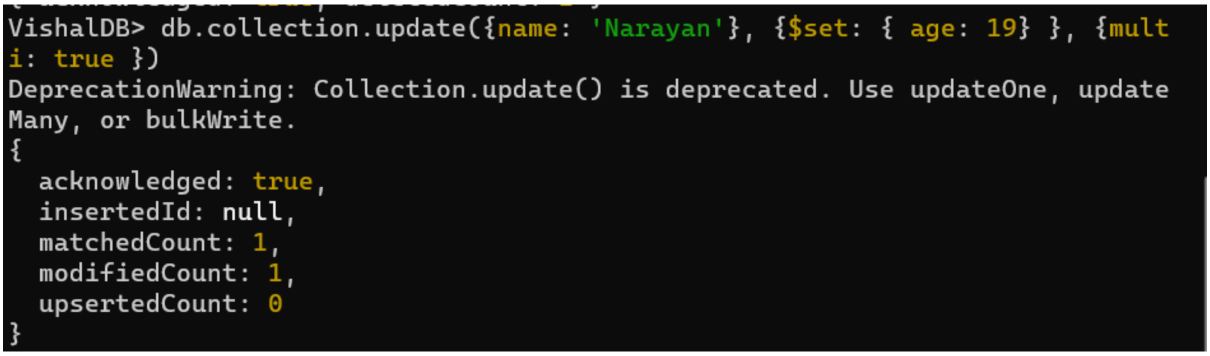
d. **remove ()**



```
VishalDB> db.collection.remove({name: 'Hina'})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, delete
Many, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
```

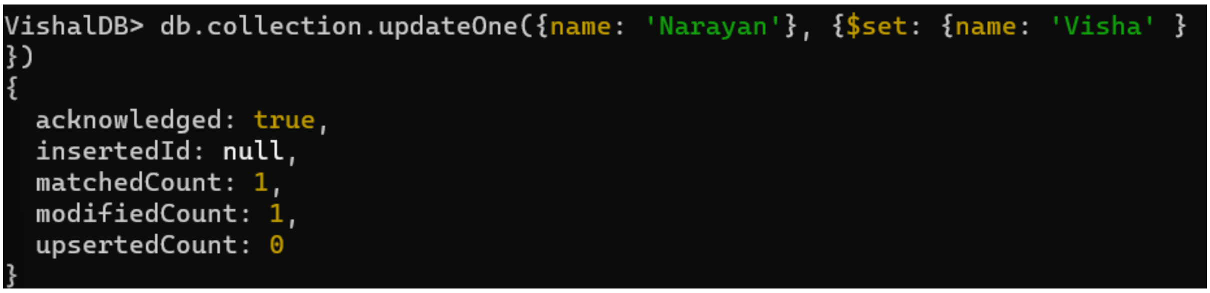
5) Write the syntax of update ( ) command and implement the following functions-

a. **Update ()**



```
VishalDB> db.collection.update({name: 'Narayan'}, {$set: { age: 19} }, {mult
i: true })
DeprecationWarning: Collection.update() is deprecated. Use updateOne, update
Many, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

b. **updateOne ()**

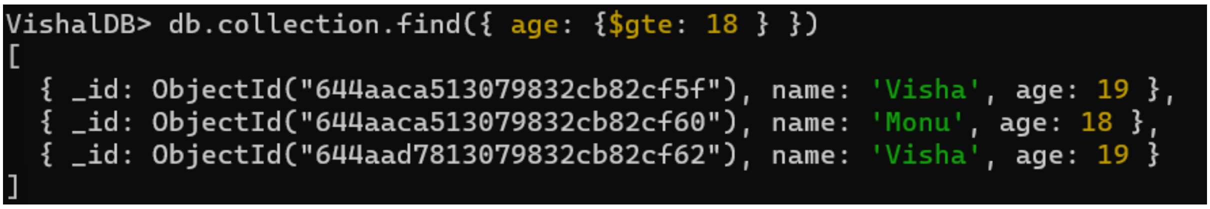


```
VishalDB> db.collection.updateOne({name: 'Narayan'}, {$set: {name: 'Visha' }
})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

6) Implement the find ( ) command and use the following operators -

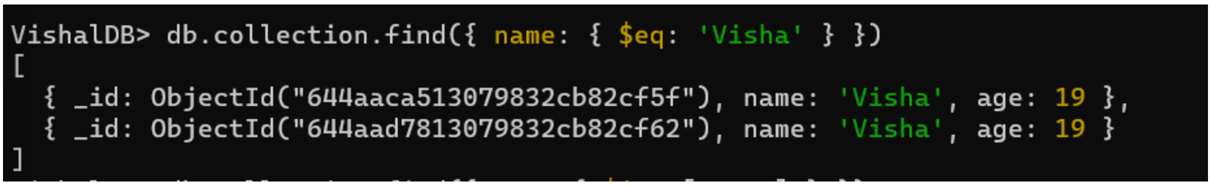
a. **Query Operators**

1. **$gte**



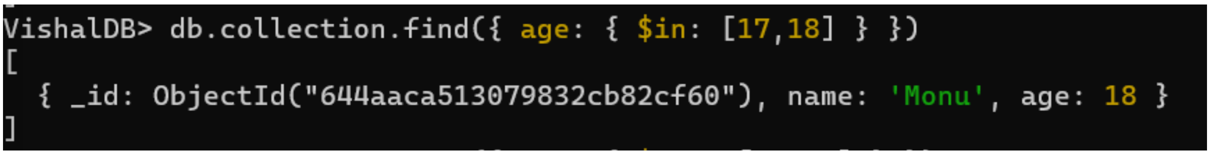
```
VishalDB> db.collection.find({ age: {$gte: 18 } })
[
  { _id: ObjectId("644aaca513079832cb82cf5f"), name: 'Visha', age: 19 },
  { _id: ObjectId("644aaca513079832cb82cf60"), name: 'Monu', age: 18 },
  { _id: ObjectId("644aad7813079832cb82cf62"), name: 'Visha', age: 19 }
]
```

2. $eq



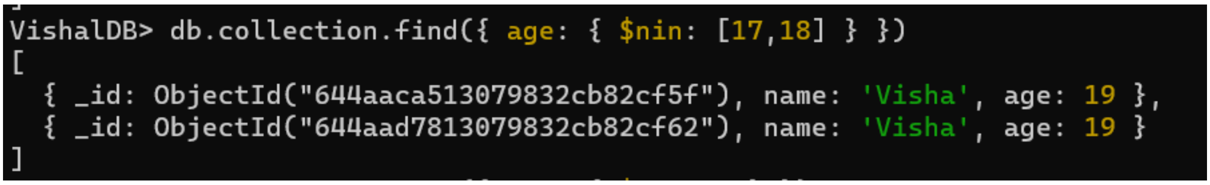
```
VishalDB> db.collection.find({ name: { $eq: 'Visha' } })
[
  { _id: ObjectId("644aaca513079832cb82cf5f"), name: 'Visha', age: 19 },
  { _id: ObjectId("644aad7813079832cb82cf62"), name: 'Visha', age: 19 }
]
```

3. $in



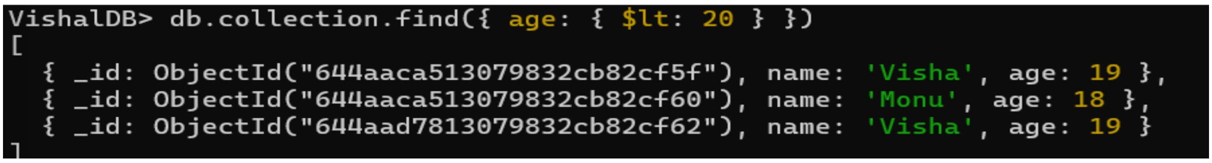
```
VishalDB> db.collection.find({ age: { $in: [17,18] } })
[
  { _id: ObjectId("644aaca513079832cb82cf60"), name: 'Monu', age: 18 }
]
```

4. $nin



```
VishalDB> db.collection.find({ age: { $nin: [17,18] } })
[
  { _id: ObjectId("644aaca513079832cb82cf5f"), name: 'Visha', age: 19 },
  { _id: ObjectId("644aad7813079832cb82cf62"), name: 'Visha', age: 19 }
]
```

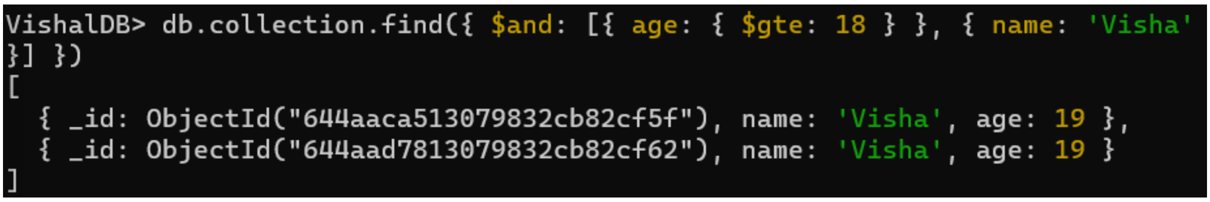
5. $lt



```
VishalDB> db.collection.find({ age: { $lt: 20 } })
[
  { _id: ObjectId("644aaca513079832cb82cf5f"), name: 'Visha', age: 19 },
  { _id: ObjectId("644aaca513079832cb82cf60"), name: 'Monu', age: 18 },
  { _id: ObjectId("644aad7813079832cb82cf62"), name: 'Visha', age: 19 }
]
```

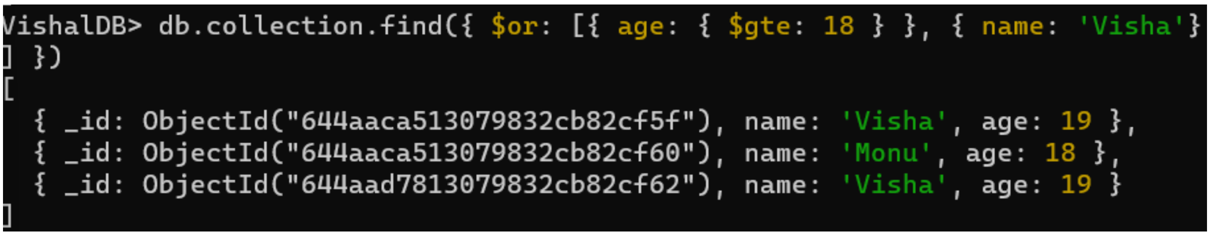
b) Logical Operators

1. $and



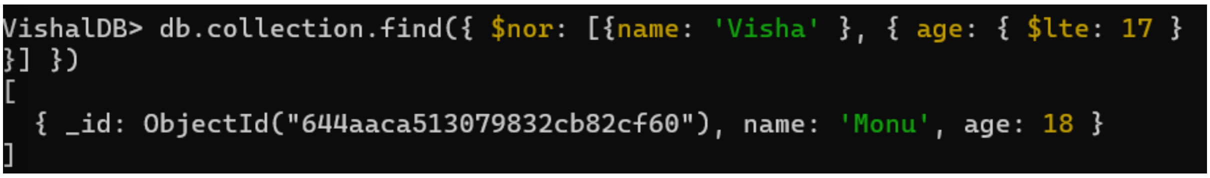
```
VishalDB> db.collection.find({ $and: [{ age: { $gte: 18 } }, { name: 'Visha'
}] })
[
  { _id: ObjectId("644aaca513079832cb82cf5f"), name: 'Visha', age: 19 },
  { _id: ObjectId("644aad7813079832cb82cf62"), name: 'Visha', age: 19 }
]
```

2. $or



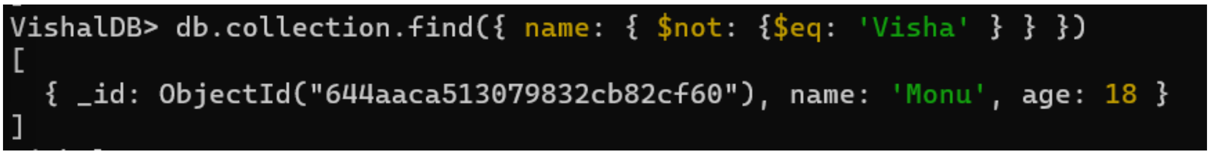
```
VishalDB> db.collection.find({ $or: [{ age: { $gte: 18 } }, { name: 'Visha'}
] })
[
  { _id: ObjectId("644aaca513079832cb82cf5f"), name: 'Visha', age: 19 },
  { _id: ObjectId("644aaca513079832cb82cf60"), name: 'Monu', age: 18 },
  { _id: ObjectId("644aad7813079832cb82cf62"), name: 'Visha', age: 19 }
]
```

3. $nor



```
VishalDB> db.collection.find({ $nor: [{name: 'Visha' }, { age: { $lte: 17 }
}] })
[
  { _id: ObjectId("644aaca513079832cb82cf60"), name: 'Monu', age: 18 }
]
```

4. $not



```
VishalDB> db.collection.find({ name: { $not: {$eq: 'Visha' } } })
[
  { _id: ObjectId("644aaca513079832cb82cf60"), name: 'Monu', age: 18 }
]
```