```
show dbs
```

```
use bank_info
```

```
db.createCollection("deposite")
```

```
db.deposite.insertOne({
    "ACTNO": 101,
    "CNAME": "ANIL",
    "BNAME": "VRCE",
    "AMOUNT": 1000,
    "ADATE": "1-3-95"
  })
```

```
db.deposite.insertMany([{
    "ACTNO": 102,
    "CNAME": "SUNIL",
    "BNAME": "AJNI",
    "AMOUNT": 5000,
    "ADATE": "4-1-96"
  },
 {
    "ACTNO": 103,
    "CNAME": "MEHUL",
    "BNAME": "KAROLBAGH",
    "AMOUNT": 3500,
    "ADATE": "17-11-95"
  },
{
    "ACTNO": 104,
    "CNAME": "MADHURI",
    "BNAME": "CHANDI",
    "AMOUNT": 1200,
    "ADATE": "17-12-95"
  },
 {
    "ACTNO": 105,
    "CNAME": "PRMOD",
    "BNAME": "M.G. ROAD",
    "AMOUNT": 3000,
    "ADATE": "27-3-96"
  },
 {
    "ACTNO": 106,
    "CNAME": "SANDIP",
    "BNAME": "ANDHERI",
    "AMOUNT": 2000,
    "ADATE": "31-3-96"
  },
{
    "ACTNO": 107,
    "CNAME": "SHIVANI",
    "BNAME": "VIRAR",
    "AMOUNT": 1000,
    "ADATE": "5-9-95"
  },
{
    "ACTNO": 108,
    "CNAME": "KRANTI",
    "BNAME": "NEHRU PLACE",
    "AMOUNT": 5000,
    "ADATE": "2-7-95"
  }])
```

Retrieve/Display every document of your collections.

```
db.deposite.find()
```

2. Retrieve/Display every document of your collection. (Use option pretty)

```
db.deposite.find().pretty()
```

3. Display only one documents of your collection. (Use findone)

```
db.deposite.findOne()
```

4. Display documents whose Account Number is 101.

```
db.deposite.find({ACTNO:101})
```

or

```
db.deposite.find({ACTNO:{$eq:101}})
```

not equal

```
db.deposite.find({ACTNO:{$ne:101}})
```

greater than

```
db.deposite.find({ACTNO:{$gt:101}})
```

greater than equal

```
db.deposite.find({ACTNO:{$gte:101}})
```

less then

```
db.deposite.find({ACTNO:{$lt:101}})
```

less than equal

```
db.deposite.find({ACTNO:{$lte:101}})
```

in operator

```
db.deposite.find({ACTNO:{$in:[101,105,107]}})
```

not in operator

```
db.deposite.find({ACTNO:{$nin:[101,105,107]}})
```

count no of record

```
db.deposite.find().count()
```

5. Display documents whose Account Number is less than 103.

```
db.deposite.find({ACTNO:{$ne:103}})
```

6. Display documents whose Account Number is greater than 102 and Customer Name is Arjun.

```
db.deposite.find({$and:[{ACTNO:{$gt:102}},{CNAME:"ARJUN"}]})
```

```
db.deposite.find({$or:[{ACTNO:{$gt:102}},{CNAME:"ARJUN"}]})
```

7. Display documents whose Account Number is 105 or 108 using IN.

```
db.deposite.find({ ACTNO :{$in:[105,108]}})
```

8. Display documents whose Account Number is not greater than 105.

```
db.deposite.find( { ACTNO: { $not: { $gt: 105 } } } )
```

or

```
db.deposite.find( { ACTNO:  { $lte: 105 }   } )
```

9. Display documents with CNAME, BNAME and AMOUNT fields.

```
db.deposite.find({},{CNAME:1,BNAME:1,AMOUNT:1})
```

10. Display Nagpur city branch’s documents with CNAME, BNAME and AMOUNT fields.

```
db.deposite.find({BNAME:"nagpur"},{CNAME:1,BNAME:1,AMOUNT:1})
```

```
//display with AutoID
```

```
db.deposite.find({BNAME:"VIRAR"},{CNAME:1,BNAME:1,AMOUNT:1})
```

//display without AUTOID

```
db.deposite.find({BNAME:"VIRAR"},{_id:0,CNAME:1,BNAME:1,AMOUNT:1})
```

11. Display every document of your collection on ascending order by CNAME and descending order by AMOUNT.

ascending order by CNAME

```
db.deposite.find().sort({CNAME:1})
```

descending order by AMOUNT.

```
db.deposite.find().sort({AMOUNT:-1})
```

12. Display only two documents of your collection.

```
db.deposite.find().limit(2)
```

13. Display from 3rd documents of your collection

```
db.deposite.find().skip(2).limit(1)
```

14. Display the count of documents in your collection.

```
db.deposite.find().count()
```

15. Display the documents whose name starts with S in your collection.

```
db.deposite.find({CNAME:/^S/i})
```

or

```
db.deposite.find({CNAME:/^S/})
```

M anywhere in name

```
db.deposite.find({CNAME:/M/})
```

name contains NI

```
db.deposite.find({CNAME:/NI/})
```

name ends with L

```
db.deposite.find({CNAME:/L$/})
```

16. Display the documents whose name starts with S or M in your collection.

```
db.deposite.find({CNAME: /^[S,M]/i})
```

17. Display the documents whose name starts with A and having 5 characters in your collection.

```
db.deposite.find({CNAME: /^A..../})
```

18. Display the documents whose name starts with A to M in your collection.

```
db.deposite.find({CNAME: /^[A-M]/i})
```

19. Display the sum of amount in your collection.

```
db.deposite.aggregate([{$group:{ _id: null , total: { $sum: "$AMOUNT" }}}])
```

20. Display branch wise sum of amount in your collection. (Use Deposite Collection)

```
db.deposite.aggregate([{$group:{ _id:"$BNAME",total:{$sum: "$AMOUNT"}}}])
```

max amount

```
db.deposite.aggregate([{$group:{ _id: null , max: { $max: "$AMOUNT" }}}])
```

min amount

```
db.deposite.aggregate([{$group:{ _id: null , min: { $min: "$AMOUNT" }}}])
```

avg amount

```
db.deposite.aggregate([{$group:{ _id: null , avg: { $avg: "$AMOUNT" }}}])
```

first amount

```
db.deposite.aggregate([{$group:{ _id: null , first: { $first: "$AMOUNT"
}}}])
```

last amount

```
db.deposite.aggregate([{$group:{ _id: null , last: { $last: "$AMOUNT" }}}])
```

21. Update name of Anil to Arjun and also Branch Name to “DPS”.

```
db.deposite.updateMany({CNAME: "ANIL"},{$set: {CNAME:"ARJUN", BNAME:
"DPS"}})
```

22. Delete the document whose Branch Name is DPS.

```
db.deposite.deleteOne({BNAME: " DPS"})
```

23. Delete the collection deposite.

```
db.deposite.drop()
```

24. Drop BANK_INFO database.

```
db.dropDatabase()
```