

```
> show dbs
admin      0.000GB
config     0.000GB
local      0.000GB
practice   0.000GB
testdb     0.000GB
> use practice
switched to db practice
> show collections
bank
sales
```

```
> db.bank.find().limit(1).pretty()
{
  "_id" : ObjectId("63995917ba9ee0a78327055e"),
  "age" : 44,
  "marital" : "married",
  "balance" : 6203,
  "contact" : "cellular",
  "duration" : 58,
  "outcome" : "failure",
  "date" : "17-Nov"
}
```

#### 1. Find the number of customers

- who are aged 40 and above
- married
- and the campaign outcome was successful for them

```
> db.bank.find({$and :[{ 'age':{$gte:40}},{'marital':'married'},{'outcome':'success'}]}).count()
32
```

2. Find the median age for the given list of customers?

```
> db.bank.find().count()
825
> db.bank.find().sort({'age':1}).skip(422).limit(1).pretty()
{
  "_id" : ObjectId("63995917ba9ee0a7832706aa"),
  "age" : 40,
  "marital" : "married",
  "balance" : 2133,
  "contact" : "cellular",
  "duration" : 377,
  "outcome" : "failure",
  "date" : "06-Apr"
}
```

3. Find the number of customers who were contacted in the month of February ('Feb')?

```
> db.bank.find({'date': {$regex:/\ -Feb/}}).count()
111
```

4. For all the customers who are

- either single or divorced
- and either contacted through cellular or telephone
- find the highest balance ?

```
> db.bank.find({'$and':[{'contact': {'$in':['cellular','telephone']}},{'marital':{'$in':['single','divorced']}}]}).sort({'balance':-1}).limit(1).pretty()
{
  "_id" : ObjectId("63995917ba9ee0a7832706b7"),
  "age" : 53,
  "marital" : "divorced",
  "balance" : 8112,
  "contact" : "cellular",
  "duration" : 75,
  "outcome" : "failure",
  "date" : "15-Apr"
}
```

5. find all customers

- who are age greater than equal to 40
- who has balance > 6000

```
> db.bank.find({'$and':[{'age': {$gte: 40}},{'balance':{'$gte:6000'}}]}).count()
9
```

#### 6. lowest balance

```
> db.bank.find().sort({balance:1}).limit(1).pretty()
{
  "_id" : ObjectId("63995917ba9ee0a7832707dd"),
  "age" : 45,
  "marital" : "married",
  "balance" : -1621,
  "contact" : "cellular",
  "duration" : 8,
  "outcome" : "failure",
  "date" : "15-May"
}
```

#### 7. highest balance

```
> db.bank.find().sort({balance:-1}).limit(1).pretty()
{
  "_id" : ObjectId("63995917ba9ee0a783270598"),
  "age" : 33,
  "marital" : "married",
  "balance" : 17455,
  "contact" : "cellular",
  "duration" : 168,
  "outcome" : "failure",
  "date" : "18-Nov"
}
```

8. Group the customers based on their marital status and find the total duration for each category. Which category has the highest call duration?

```
> db.bank.aggregate([{$group:{_id:'$marital', totaldur: {$sum:'$duration'}}}])
{ "_id" : "married", "totaldur" : 132654 }
{ "_id" : "divorced", "totaldur" : 23744 }
{ "_id" : "single", "totaldur" : 43638 }
```

9. Continuing from the previous question, modify the aggregation command that you wrote to include the average balance for each category in the 'marital' field as well. Which category has the lowest average balance and which one has the lowest total duration?

```
> db.bank.aggregate([{$group:{_id:'$marital', totaldur: {$sum:'$duration'}, avgbal: {$avg:'$balance'}}}])
{ "_id" : "single", "totaldur" : 43638, "avgbal" : 587.0055555555556 }
{ "_id" : "divorced", "totaldur" : 23744, "avgbal" : 910.2717391304348 }
{ "_id" : "married", "totaldur" : 132654, "avgbal" : 1046.5678119349006 }
```