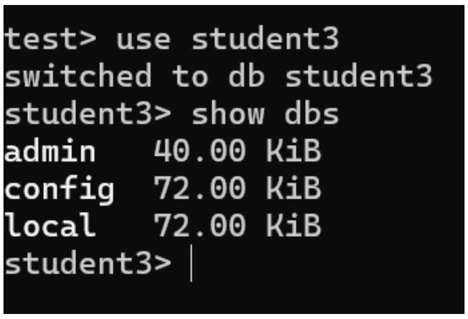
Experiment NO :-3

Titel: Implement basic database queries MangoDb

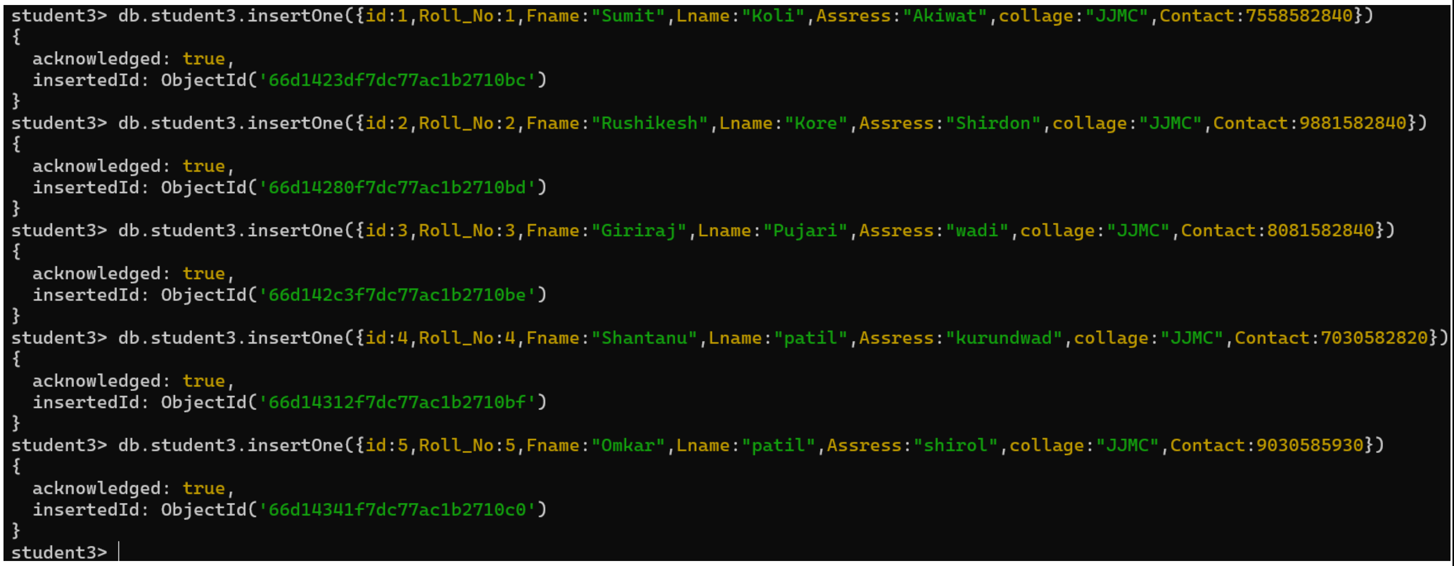
Problem Statement: Implement CRUD opration using MongoDb FOR given Set of qution



```
test> use student3
switched to db student3
student3> show dbs
admin    40.00 KiB
config   72.00 KiB
local    72.00 KiB
student3>
```

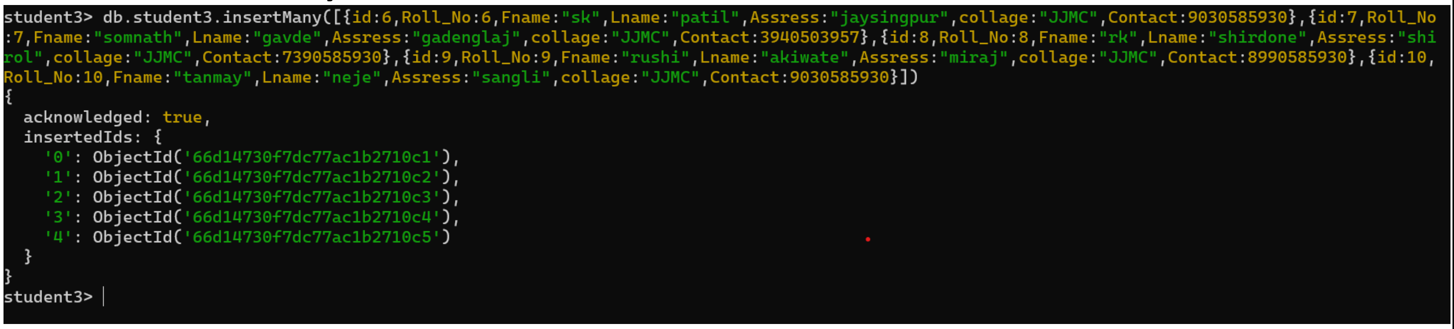
Q1). Create a database Student,Create collection student in studentDatabase and print the collection

Q2.) ınser following documention in collection StudentId ,roll_no,Fname,Lname,Adderes,college,contact



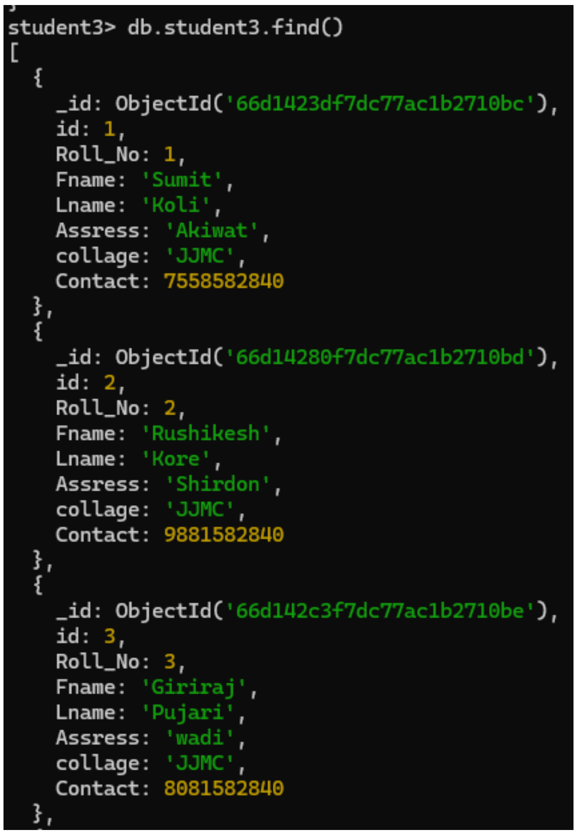
```
student3> db.student3.insertOne({id:1,Roll_No:1,Fname:"Sumit",Lname:"Koli",Assress:"Akiwat",collage:"JJMC",Contact:7558582840})
{
  acknowledged: true,
  insertedId: ObjectId('66d1423df7dc77ac1b2710bc')
}
student3> db.student3.insertOne({id:2,Roll_No:2,Fname:"Rushikesh",Lname:"Kore",Assress:"Shirdon",collage:"JJMC",Contact:9881582840})
{
  acknowledged: true,
  insertedId: ObjectId('66d14280f7dc77ac1b2710bd')
}
student3> db.student3.insertOne({id:3,Roll_No:3,Fname:"Giriraj",Lname:"Pujari",Assress:"wadi",collage:"JJMC",Contact:8081582840})
{
  acknowledged: true,
  insertedId: ObjectId('66d142c3f7dc77ac1b2710be')
}
student3> db.student3.insertOne({id:4,Roll_No:4,Fname:"Shantanu",Lname:"patil",Assress:"kurundwad",collage:"JJMC",Contact:7030582820})
{
  acknowledged: true,
  insertedId: ObjectId('66d14312f7dc77ac1b2710bf')
}
student3> db.student3.insertOne({id:5,Roll_No:5,Fname:"Omkar",Lname:"patil",Assress:"shirol",collage:"JJMC",Contact:9030585930})
{
  acknowledged: true,
  insertedId: ObjectId('66d14341f7dc77ac1b2710c0')
}
student3>
```

InsertMany:

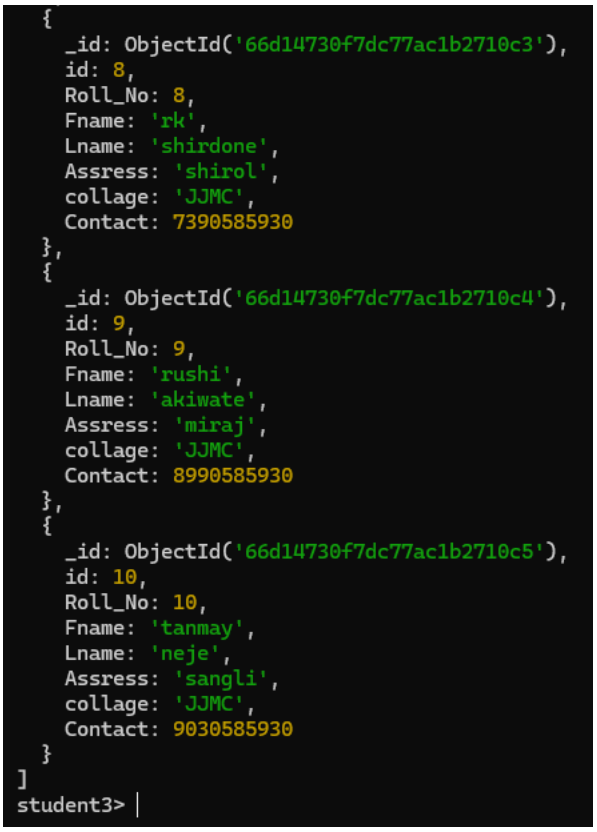


```
student3> db.student3.insertMany([{id:6,Roll_No:6,Fname:"sk",Lname:"patil",Assress:"jaysingpur",collage:"JJMC",Contact:9030585930},{id:7,Roll_No
:7,Fname:"somnath",Lname:"gavde",Assress:"gadenglaj",collage:"JJMC",Contact:3940503957},{id:8,Roll_No:8,Fname:"rk",Lname:"shirdone",Assress:"shi
rol",collage:"JJMC",Contact:7390585930},{id:9,Roll_No:9,Fname:"rushi",Lname:"akiwate",Assress:"miraj",collage:"JJMC",Contact:8990585930},{id:10,
Roll_No:10,Fname:"tanmay",Lname:"neje",Assress:"sangli",collage:"JJMC",Contact:9030585930}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66d14730f7dc77ac1b2710c1'),
    '1': ObjectId('66d14730f7dc77ac1b2710c2'),
    '2': ObjectId('66d14730f7dc77ac1b2710c3'),
    '3': ObjectId('66d14730f7dc77ac1b2710c4'),
    '4': ObjectId('66d14730f7dc77ac1b2710c5')
  }
}
student3>
```

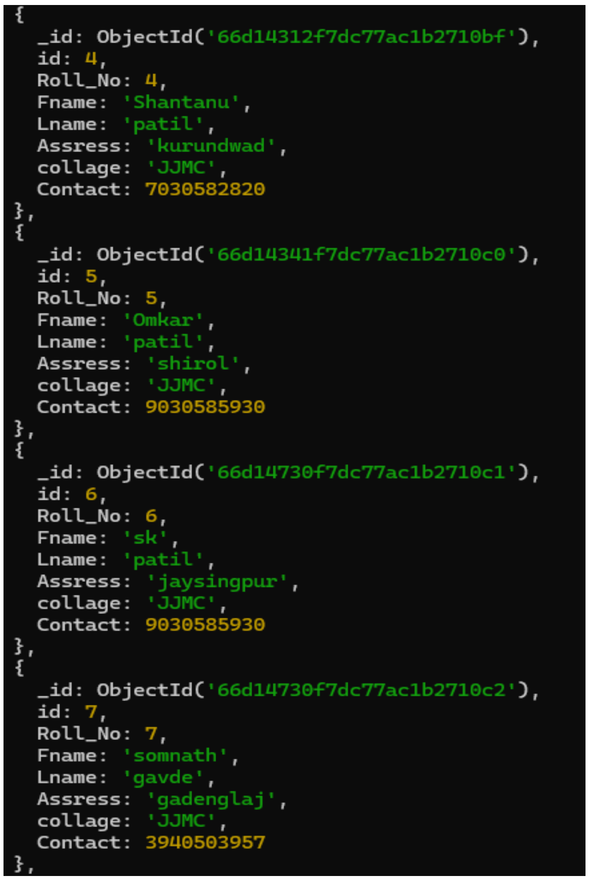
Q3.) Print all the document from collection student



```
student3> db.student3.find()
[
  {
    _id: ObjectId('66d1423df7dc77ac1b2710bc'),
    id: 1,
    Roll_No: 1,
    Fname: 'Sumit',
    Lname: 'Koli',
    Assress: 'Akiwat',
    collage: 'JJMC',
    Contact: 7558582840
  },
  {
    _id: ObjectId('66d14280f7dc77ac1b2710bd'),
    id: 2,
    Roll_No: 2,
    Fname: 'Rushikesh',
    Lname: 'Kore',
    Assress: 'Shirdon',
    collage: 'JJMC',
    Contact: 9881582840
  },
  {
    _id: ObjectId('66d142c3f7dc77ac1b2710be'),
    id: 3,
    Roll_No: 3,
    Fname: 'Giriraj',
    Lname: 'Pujari',
    Assress: 'wadi',
    collage: 'JJMC',
    Contact: 8081582840
  },
```

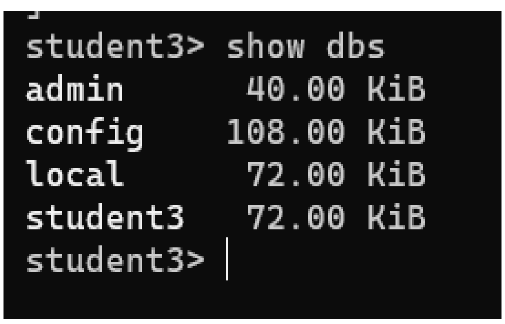


```
  {
    _id: ObjectId('66d14730f7dc77ac1b2710c3'),
    id: 8,
    Roll_No: 8,
    Fname: 'rk',
    Lname: 'shirdone',
    Assress: 'shirol',
    collage: 'JJMC',
    Contact: 7390585930
  },
  {
    _id: ObjectId('66d14730f7dc77ac1b2710c4'),
    id: 9,
    Roll_No: 9,
    Fname: 'rushi',
    Lname: 'akiwate',
    Assress: 'miraj',
    collage: 'JJMC',
    Contact: 8990585930
  },
  {
    _id: ObjectId('66d14730f7dc77ac1b2710c5'),
    id: 10,
    Roll_No: 10,
    Fname: 'tanmay',
    Lname: 'neje',
    Assress: 'sangli',
    collage: 'JJMC',
    Contact: 9030585930
  }
]
student3> |
```

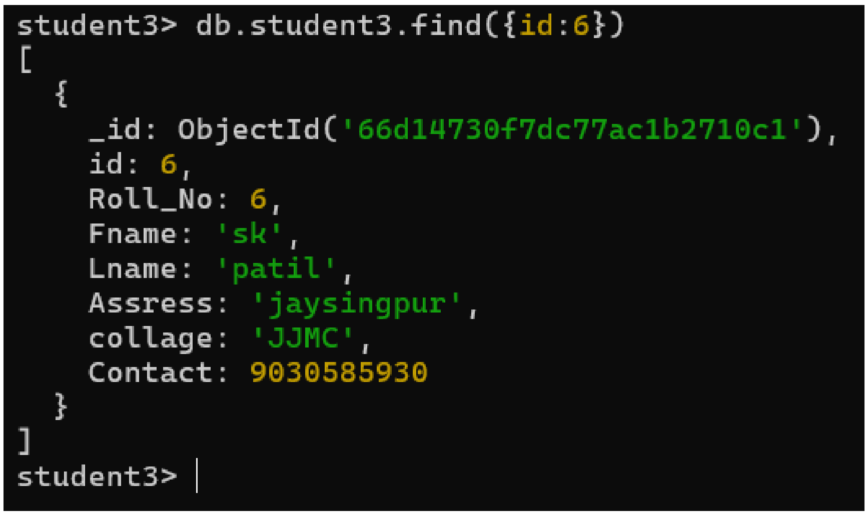


```
{
  _id: ObjectId('66d14312f7dc77ac1b2710bf'),
  id: 4,
  Roll_No: 4,
  Fname: 'Shantanu',
  Lname: 'patil',
  Assress: 'kurundwad',
  collage: 'JJMC',
  Contact: 7030582820
},
{
  _id: ObjectId('66d14341f7dc77ac1b2710c0'),
  id: 5,
  Roll_No: 5,
  Fname: 'Omkar',
  Lname: 'patil',
  Assress: 'shirol',
  collage: 'JJMC',
  Contact: 9030585930
},
{
  _id: ObjectId('66d14730f7dc77ac1b2710c1'),
  id: 6,
  Roll_No: 6,
  Fname: 'sk',
  Lname: 'patil',
  Assress: 'jaysingpur',
  collage: 'JJMC',
  Contact: 9030585930
},
{
  _id: ObjectId('66d14730f7dc77ac1b2710c2'),
  id: 7,
  Roll_No: 7,
  Fname: 'somnath',
  Lname: 'gavde',
  Assress: 'gadenglaj',
  collage: 'JJMC',
  Contact: 3940503957
},
```

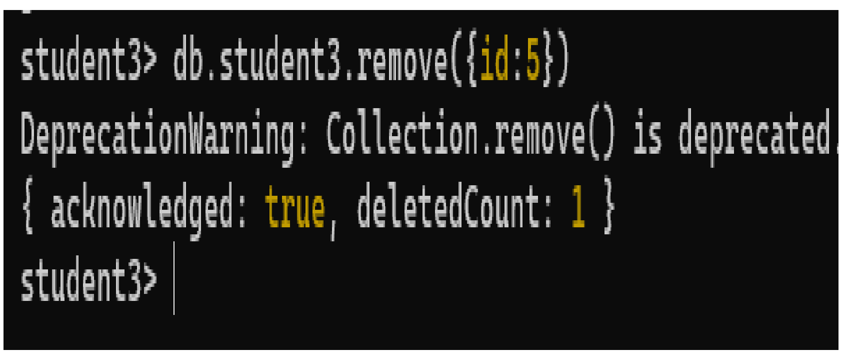
Q4.) Print all databases

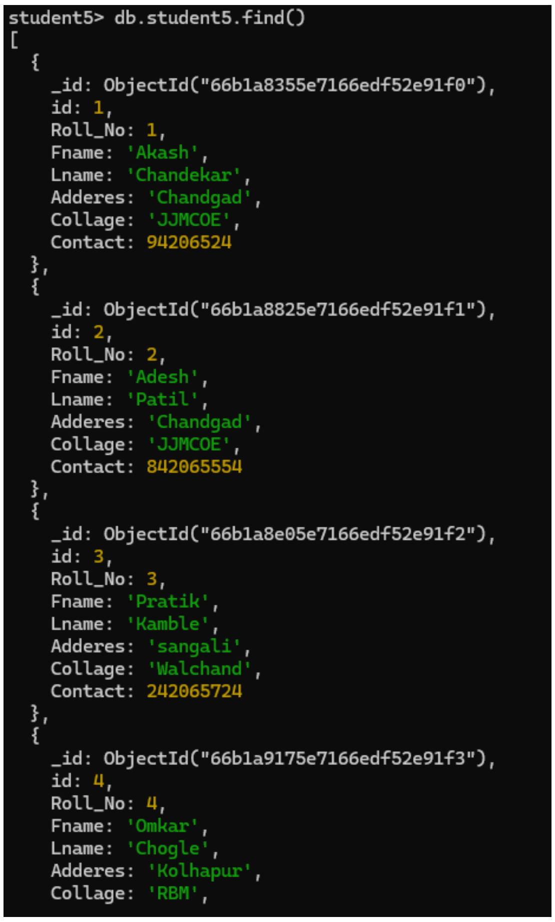

```
student3> show dbs
admin       40.00 KiB
config     108.00 KiB
local       72.00 KiB
student3    72.00 KiB
student3>
```

Q.5) Update the document of collection student with roll_no:6 where id:6


```
student3> db.student3.find({id:6})
[
  {
    _id: ObjectId('66d14730f7dc77ac1b2710c1'),
    id: 6,
    Roll_No: 6,
    Fname: 'sk',
    Lname: 'patil',
    Assress: 'jaysingpur',
    collage: 'JJMC',
    Contact: 9030585930
  }
]
student3>
```

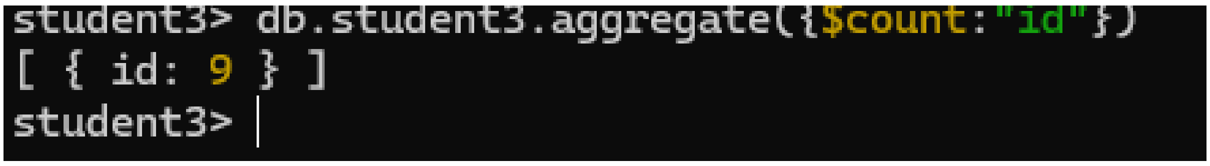
Q.6) Delete the document of collection student **id:5**


```
student3> db.student3.remove({id:5})
DeprecationWarning: Collection.remove() is deprecated
{ acknowledged: true, deletedCount: 1 }
student3>
```



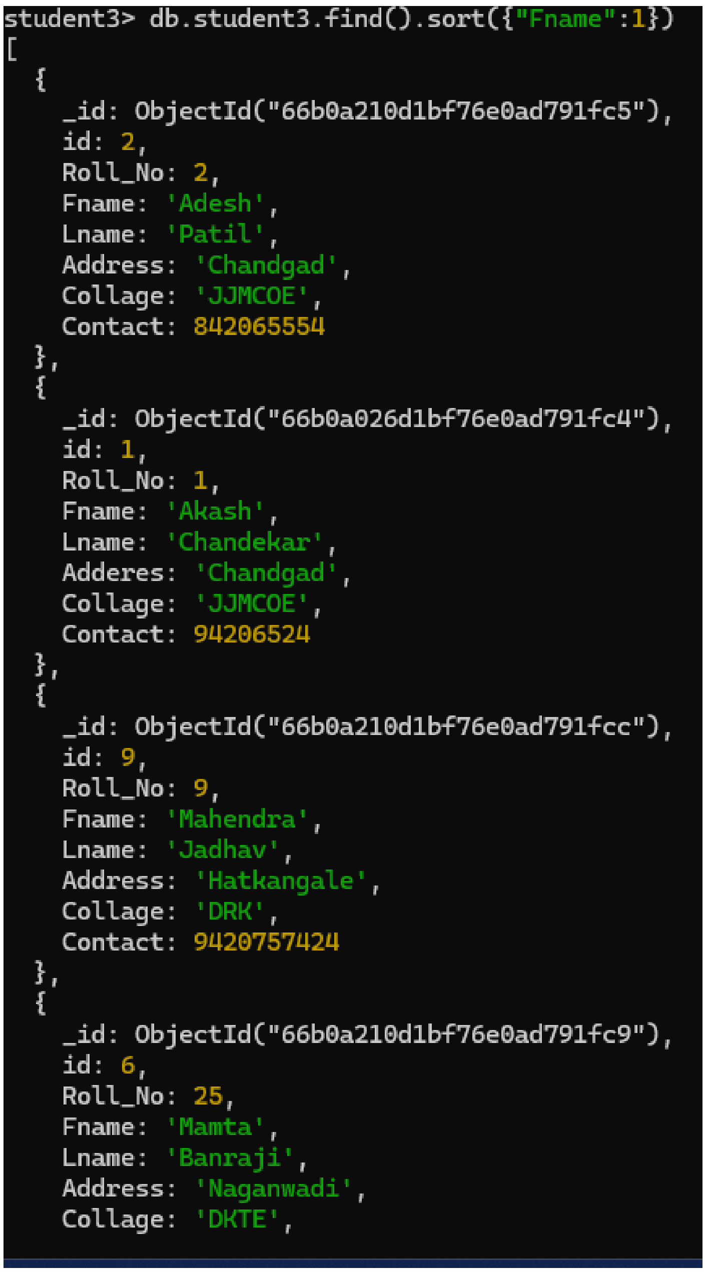
```
student5> db.student5.find()
[
  {
    _id: ObjectId("66b1a8355e7166edf52e91f0"),
    id: 1,
    Roll_No: 1,
    Fname: 'Akash',
    Lname: 'Chandekar',
    Adderes: 'Chandgad',
    Collage: 'JJMCOE',
    Contact: 94206524
  },
  {
    _id: ObjectId("66b1a8825e7166edf52e91f1"),
    id: 2,
    Roll_No: 2,
    Fname: 'Adesh',
    Lname: 'Patil',
    Adderes: 'Chandgad',
    Collage: 'JJMCOE',
    Contact: 842065554
  },
  {
    _id: ObjectId("66b1a8e05e7166edf52e91f2"),
    id: 3,
    Roll_No: 3,
    Fname: 'Pratik',
    Lname: 'Kamble',
    Adderes: 'sangali',
    Collage: 'Walchand',
    Contact: 242065724
  },
  {
    _id: ObjectId("66b1a9175e7166edf52e91f3"),
    id: 4,
    Roll_No: 4,
    Fname: 'Omkar',
    Lname: 'Chogle',
    Adderes: 'Kolhapur',
    Collage: 'RBM',
```

Q.7) count all the document from a student collection

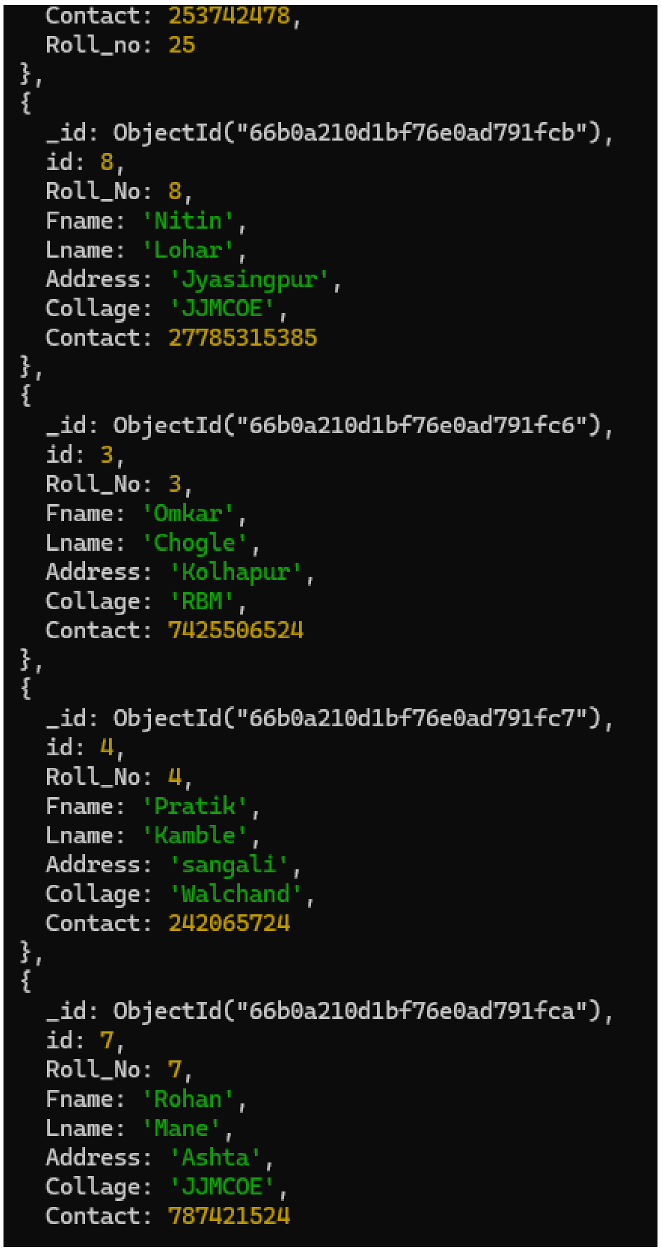


```
student3> db.student3.aggregate({$count:"id"})
[ { id: 9 } ]
student3>
```

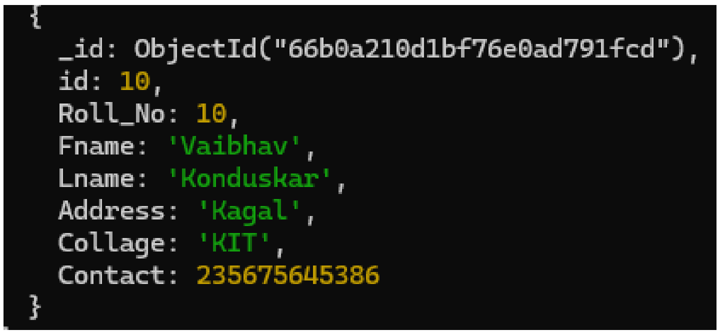
Q.8) Sort the document from a student collection order of Fname



```
student3> db.student3.find().sort({"Fname":1})
[
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fc5"),
    id: 2,
    Roll_No: 2,
    Fname: 'Adesh',
    Lname: 'Patil',
    Address: 'Chandgad',
    Collage: 'JJMCOE',
    Contact: 842065554
  },
  {
    _id: ObjectId("66b0a026d1bf76e0ad791fc4"),
    id: 1,
    Roll_No: 1,
    Fname: 'Akash',
    Lname: 'Chandekar',
    Adderes: 'Chandgad',
    Collage: 'JJMCOE',
    Contact: 94206524
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fcc"),
    id: 9,
    Roll_No: 9,
    Fname: 'Mahendra',
    Lname: 'Jadhav',
    Address: 'Hatkangale',
    Collage: 'DRK',
    Contact: 9420757424
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fc9"),
    id: 6,
    Roll_No: 25,
    Fname: 'Mamta',
    Lname: 'Banraji',
    Address: 'Naganwadi',
    Collage: 'DKTE',
```

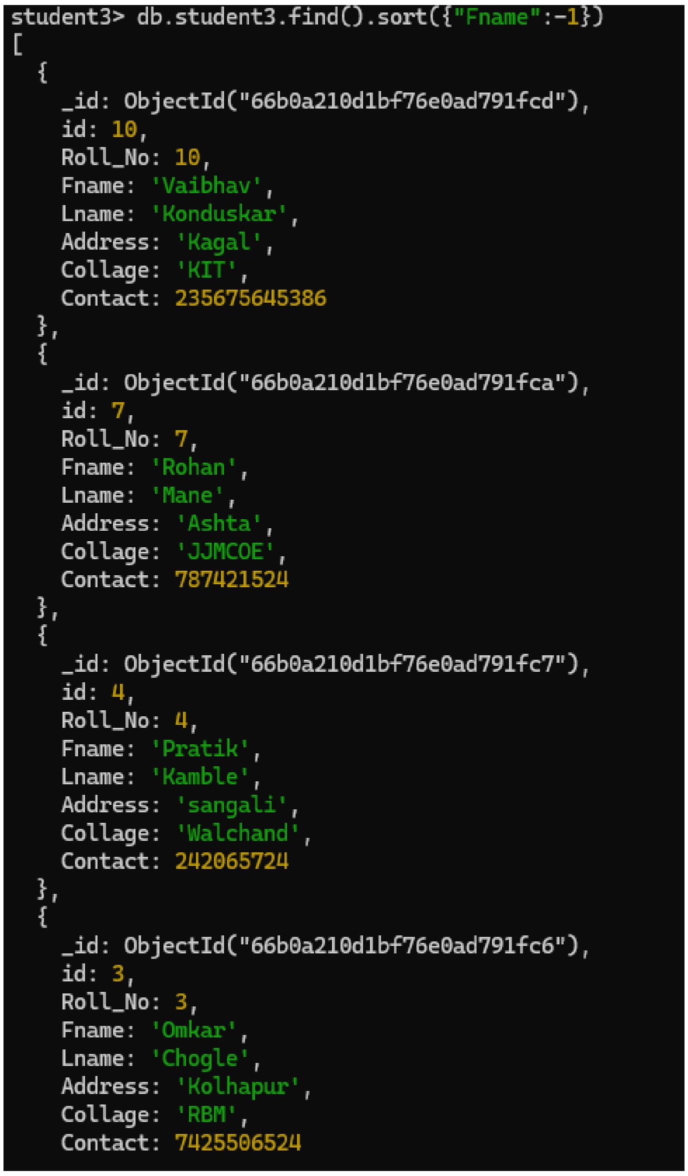


```
    Contact: 253742478,
    Roll_no: 25
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fcb"),
    id: 8,
    Roll_No: 8,
    Fname: 'Nitin',
    Lname: 'Lohar',
    Address: 'Jyasingpur',
    Collage: 'JJMCOE',
    Contact: 27785315385
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fc6"),
    id: 3,
    Roll_No: 3,
    Fname: 'Omkar',
    Lname: 'Chogle',
    Address: 'Kolhapur',
    Collage: 'RBM',
    Contact: 7425506524
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fc7"),
    id: 4,
    Roll_No: 4,
    Fname: 'Pratik',
    Lname: 'Kamble',
    Address: 'sangali',
    Collage: 'Walchand',
    Contact: 242065724
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fca"),
    id: 7,
    Roll_No: 7,
    Fname: 'Rohan',
    Lname: 'Mane',
    Address: 'Ashta',
    Collage: 'JJMCOE',
    Contact: 787421524
```

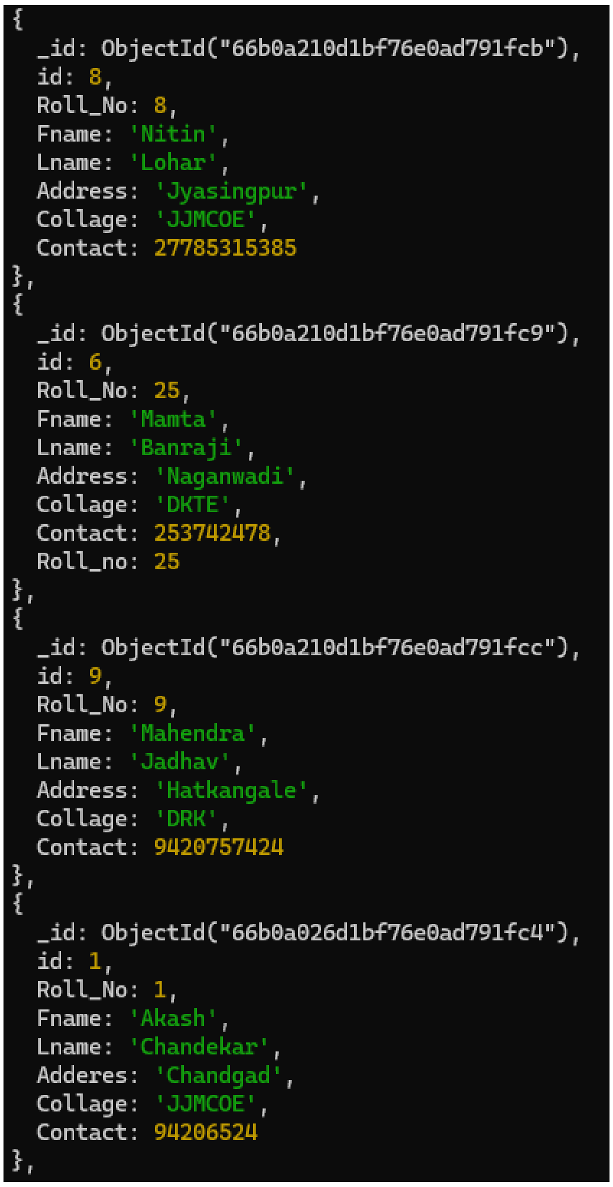


```
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fcd"),
    id: 10,
    Roll_No: 10,
    Fname: 'Vaibhav',
    Lname: 'Konduskar',
    Address: 'Kagal',
    Collage: 'KIT',
    Contact: 235675645386
  }
```

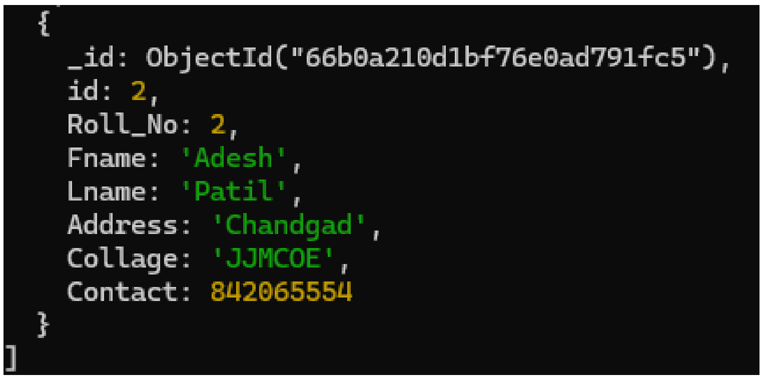
Q.8) Sort the document from a student collection in descending order of Fname



```
student3> db.student3.find().sort({"Fname":-1})
[
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fcd"),
    id: 10,
    Roll_No: 10,
    Fname: 'Vaibhav',
    Lname: 'Konduskar',
    Address: 'Kagal',
    Collage: 'KIT',
    Contact: 235675645386
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fca"),
    id: 7,
    Roll_No: 7,
    Fname: 'Rohan',
    Lname: 'Mane',
    Address: 'Ashta',
    Collage: 'JJMCOE',
    Contact: 787421524
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fc7"),
    id: 4,
    Roll_No: 4,
    Fname: 'Pratik',
    Lname: 'Kamble',
    Address: 'sangali',
    Collage: 'Walchand',
    Contact: 242065724
  },
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fc6"),
    id: 3,
    Roll_No: 3,
    Fname: 'Omkar',
    Lname: 'Chogle',
    Address: 'Kolhapur',
    Collage: 'RBM',
    Contact: 7425506524
```



```
{
  _id: ObjectId("66b0a210d1bf76e0ad791fcb"),
  id: 8,
  Roll_No: 8,
  Fname: 'Nitin',
  Lname: 'Lohar',
  Address: 'Jyasingpur',
  Collage: 'JJMCOE',
  Contact: 27785315385
},
{
  _id: ObjectId("66b0a210d1bf76e0ad791fc9"),
  id: 6,
  Roll_No: 25,
  Fname: 'Mamta',
  Lname: 'Banraji',
  Address: 'Naganwadi',
  Collage: 'DKTE',
  Contact: 253742478,
  Roll_no: 25
},
{
  _id: ObjectId("66b0a210d1bf76e0ad791fcc"),
  id: 9,
  Roll_No: 9,
  Fname: 'Mahendra',
  Lname: 'Jadhav',
  Address: 'Hatkangale',
  Collage: 'DRK',
  Contact: 9420757424
},
{
  _id: ObjectId("66b0a026d1bf76e0ad791fc4"),
  id: 1,
  Roll_No: 1,
  Fname: 'Akash',
  Lname: 'Chandekar',
  Adderes: 'Chandgad',
  Collage: 'JJMCOE',
  Contact: 94206524
},
```



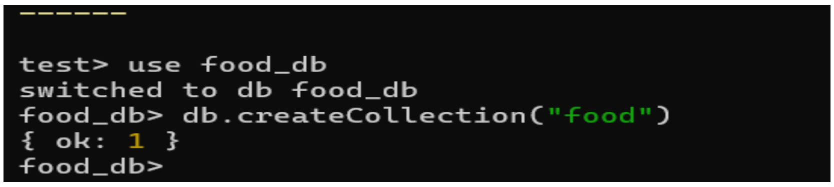
```
  {
    _id: ObjectId("66b0a210d1bf76e0ad791fc5"),
    id: 2,
    Roll_No: 2,
    Fname: 'Adesh',
    Lname: 'Patil',
    Address: 'Chandgad',
    Collage: 'JJMCOE',
    Contact: 842065554
  }
]
```

Experiment No :4

Title: Implement Basic Databases queries using MongoDB.

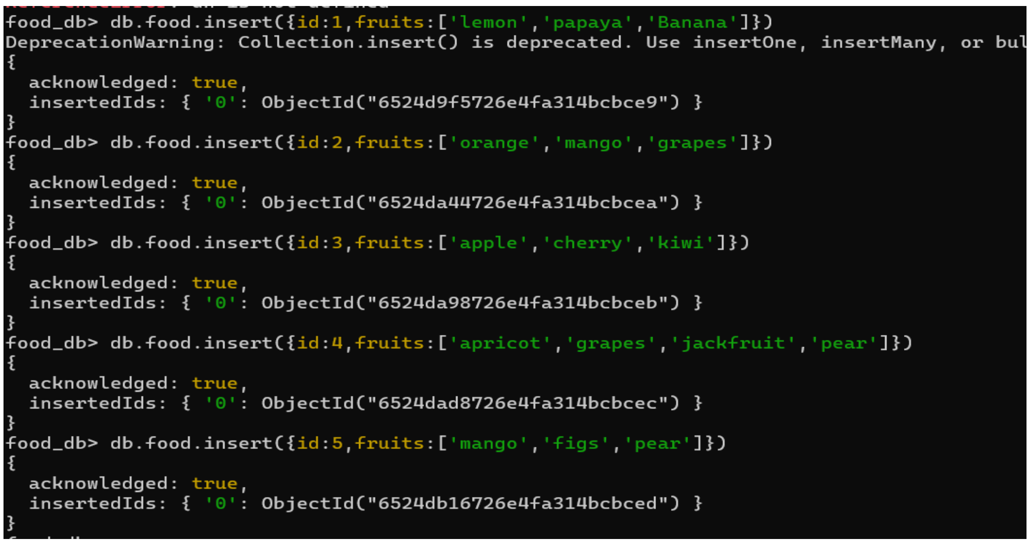
Aim: Implement arrays in MongoDB for a given set of questions.

Q1. create database food_db ,create collection food in food_db and print the collection.



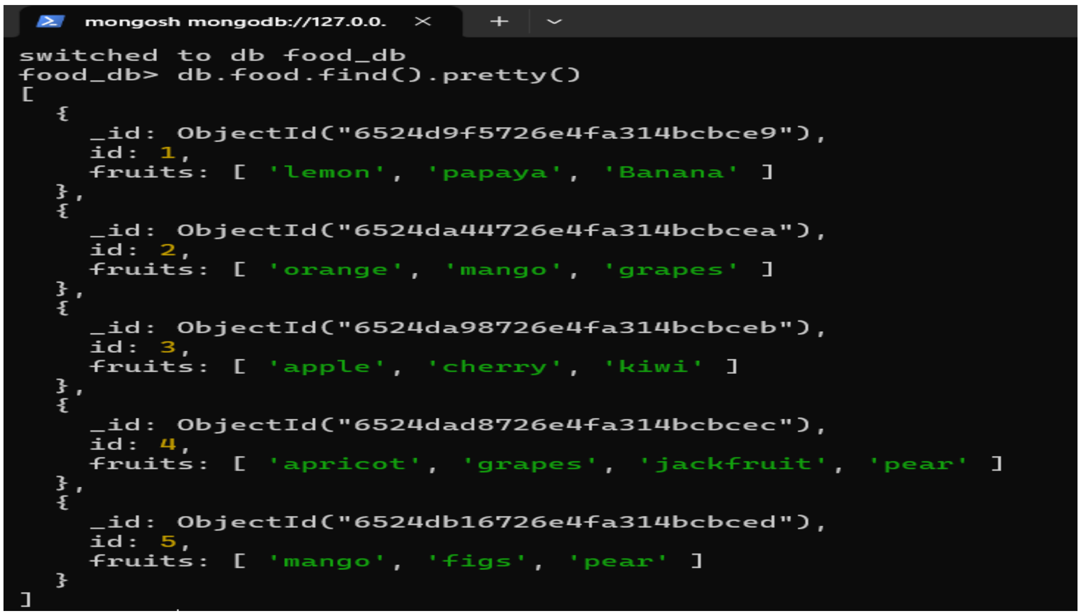
```
-------

test> use food_db
switched to db food_db
food_db> db.createCollection("food")
{ ok: 1 }
food_db>
```

Q2.Insert Following Document collection food (id,fruit)



```
food_db> db.food.insert({id:1,fruits:['lemon','papaya','Banana']})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bul
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6524d9f5726e4fa314bcbce9") }
}
food_db> db.food.insert({id:2,fruits:['orange','mango','grapes']})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6524da44726e4fa314bcbcea") }
}
food_db> db.food.insert({id:3,fruits:['apple','cherry','kiwi']})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6524da98726e4fa314bcbceb") }
}
food_db> db.food.insert({id:4,fruits:['apricot','grapes','jackfruit','pear']})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6524dad8726e4fa314bcbcec") }
}
food_db> db.food.insert({id:5,fruits:['mango','figs','pear']})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6524db16726e4fa314bcbced") }
}
```

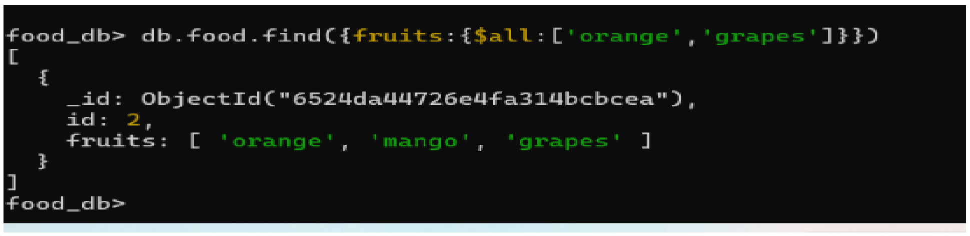
Q3.Print all the documents from collection food in a formatted manner.



```
switched to db food_db
food_db> db.food.find().pretty()
[
  {
    _id: ObjectId("6524d9f5726e4fa314bcbce9"),
    id: 1,
    fruits: [ 'lemon', 'papaya', 'Banana' ]
  },
  {
    _id: ObjectId("6524da44726e4fa314bcbcea"),
    id: 2,
    fruits: [ 'orange', 'mango', 'grapes' ]
  },
  {
    _id: ObjectId("6524da98726e4fa314bcbceb"),
    id: 3,
    fruits: [ 'apple', 'cherry', 'kiwi' ]
  },
  {
    _id: ObjectId("6524dad8726e4fa314bcbcec"),
    id: 4,
    fruits: [ 'apricot', 'grapes', 'jackfruit', 'pear' ]
  },
  {
    _id: ObjectId("6524db16726e4fa314bcbced"),
    id: 5,
    fruits: [ 'mango', 'figs', 'pear' ]
  }
]
```

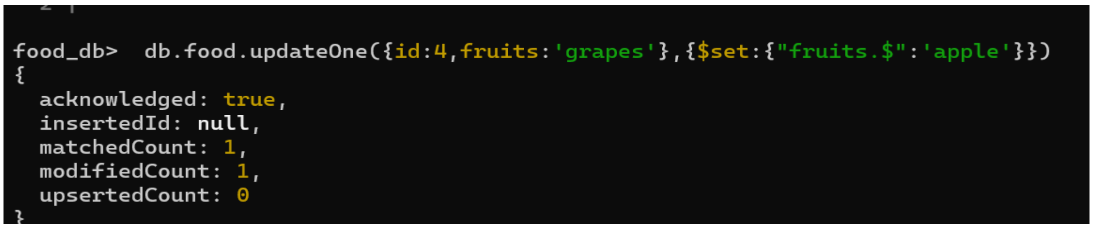
Experiment No :4

Q4.Find the documents from fruits collection which have element orange and grapes in array fruit.



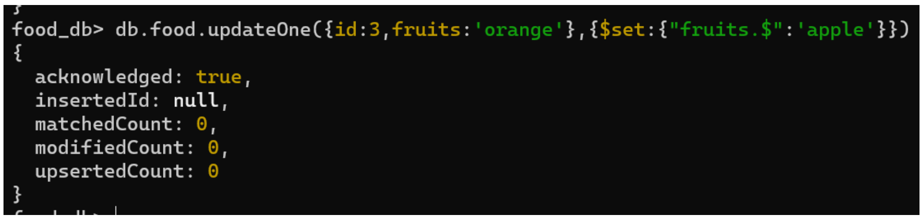
```
food_db> db.food.find({fruits:{$all:['orange','grapes']}})
[
  {
    _id: ObjectId("6524da44726e4fa314bcbcea"),
    id: 2,
    fruits: [ 'orange', 'mango', 'grapes' ]
  }
]
food_db>
```

Q5.Update the documents with id 4 & replace fruit array element with apple



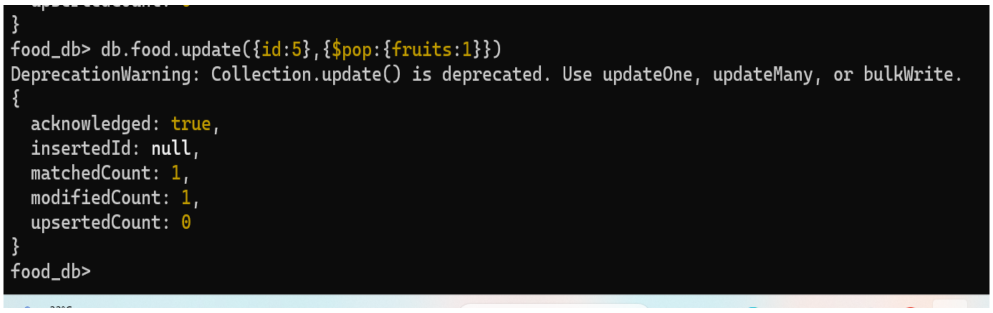
```
food_db>  db.food.updateOne({id:4,fruits:'grapes'},{$set:{"fruits.$":'apple'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Q6.Update the documents with id 3 & replace fruit array element Apple and orange.



```
food_db> db.food.updateOne({id:3,fruits:'orange'},{$set:{"fruits.$":'apple'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

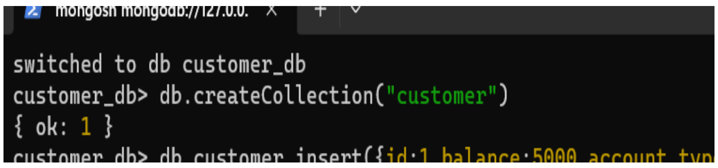
Q7.Update the documents with id 5 and remove an element from array fruits



```
}
food_db> db.food.update({id:5},{$pop:{fruits:1}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
food_db>
```

B. Implement aggregate functions for a given set of questions.

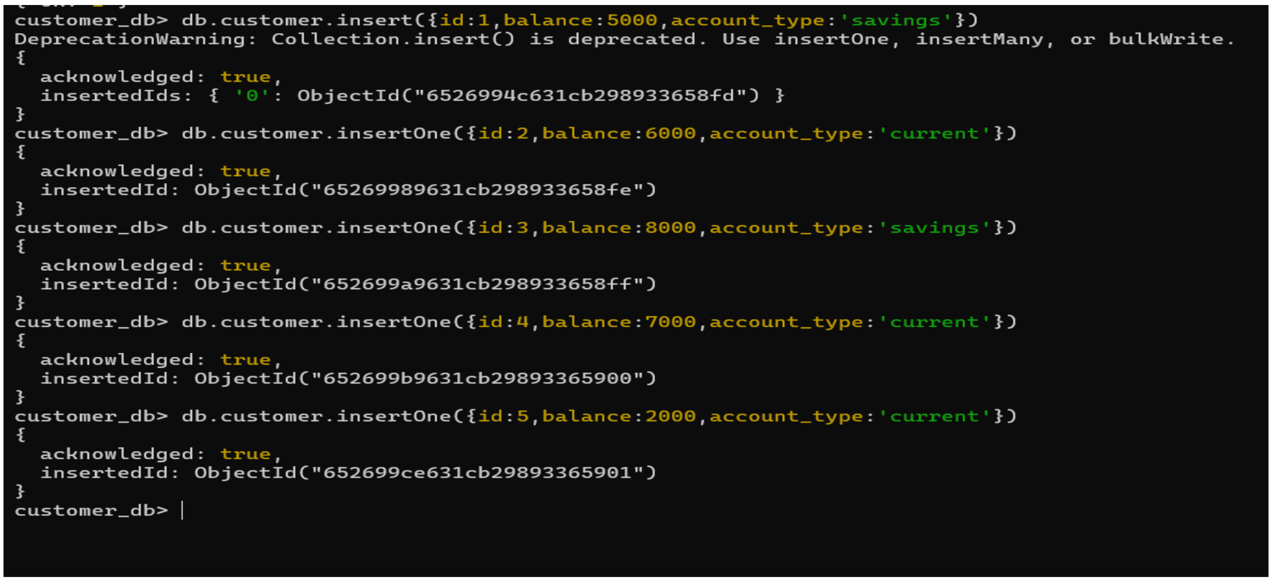
Q1.create a database customer_db,create collection customer in customer_db and print the collection



```
switched to db customer_db
customer_db> db.createCollection("customer")
{ ok: 1 }
customer_db> db.customer.insert({id:1,balance:5000,account_typ
```

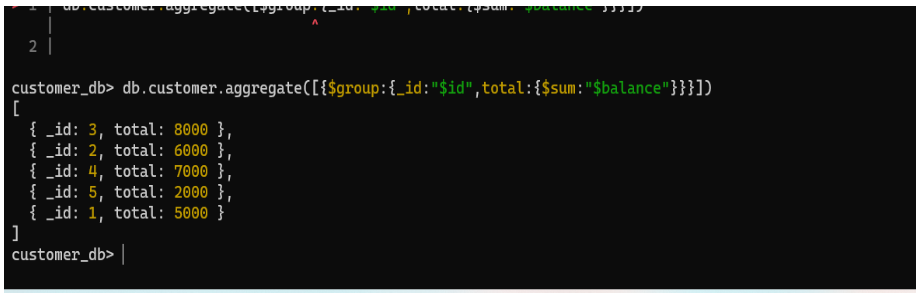
Q2.Insert Following Documents in collection customer(id,balance)

Experiment No :4



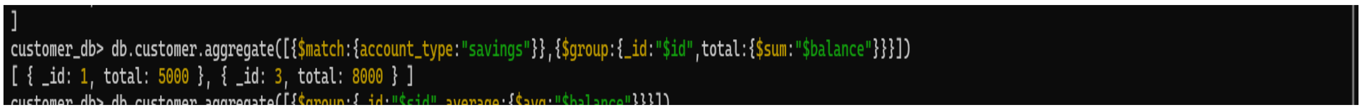
```
customer_db> db.customer.insert({id:1,balance:5000,account_type:'savings'})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("6526994c631cb298933658fd") }
}
customer_db> db.customer.insertOne({id:2,balance:6000,account_type:'current'})
{
  acknowledged: true,
  insertedId: ObjectId("65269989631cb298933658fe")
}
customer_db> db.customer.insertOne({id:3,balance:8000,account_type:'savings'})
{
  acknowledged: true,
  insertedId: ObjectId("652699a9631cb298933658ff")
}
customer_db> db.customer.insertOne({id:4,balance:7000,account_type:'current'})
{
  acknowledged: true,
  insertedId: ObjectId("652699b9631cb29893365900")
}
customer_db> db.customer.insertOne({id:5,balance:2000,account_type:'current'})
{
  acknowledged: true,
  insertedId: ObjectId("652699ce631cb29893365901")
}
customer_db> |
```

Q3. compute the sum of account balance by grouping on customer id



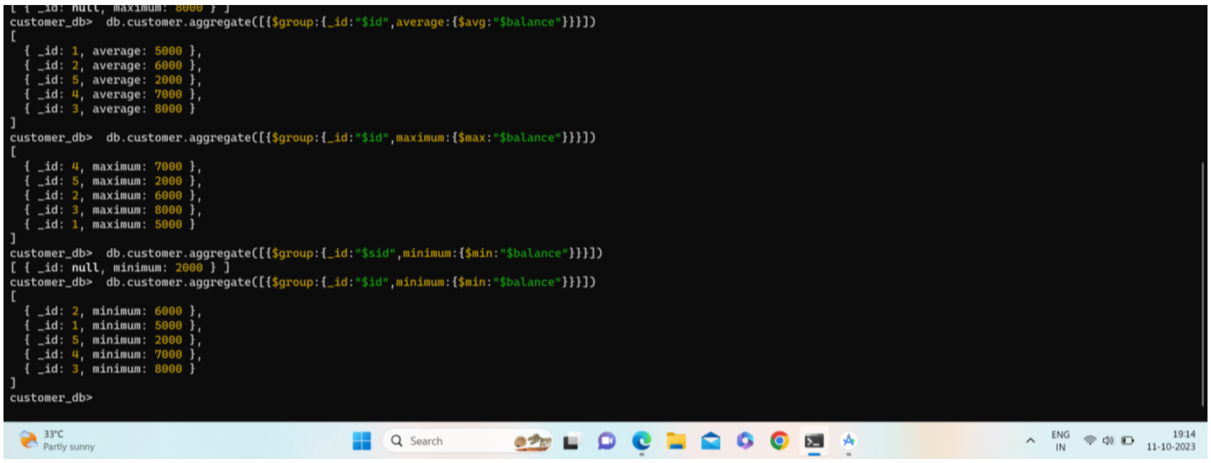
```
  2 |

customer_db> db.customer.aggregate([{$group:{_id:"$id",total:{$sum:"$balance"}}}])
[
  { _id: 3, total: 8000 },
  { _id: 2, total: 6000 },
  { _id: 4, total: 7000 },
  { _id: 5, total: 2000 },
  { _id: 1, total: 5000 }
]
customer_db> |
```

Q4.Compute the same of account balance by grouping on customer whos account type is saving account.



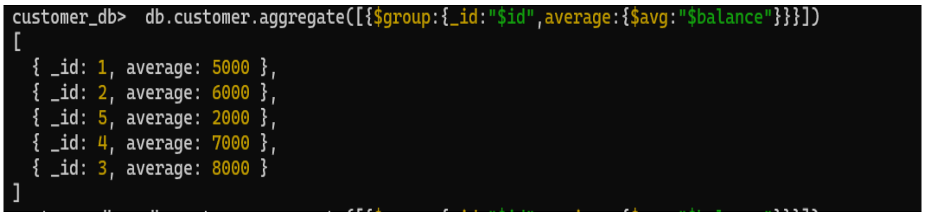
```
]
customer_db> db.customer.aggregate([{$match:{account_type:"savings"}},{$group:{_id:"$id",total:{$sum:"$balance"}}}])
[ { _id: 1, total: 5000 }, { _id: 3, total: 8000 } ]
customer_db> db.customer.aggregate([{$group:{_id:"$id",average:{$avg:"$balance"}}}])
```

Q5.compute the average of account balance by grouping on customer id for each group.



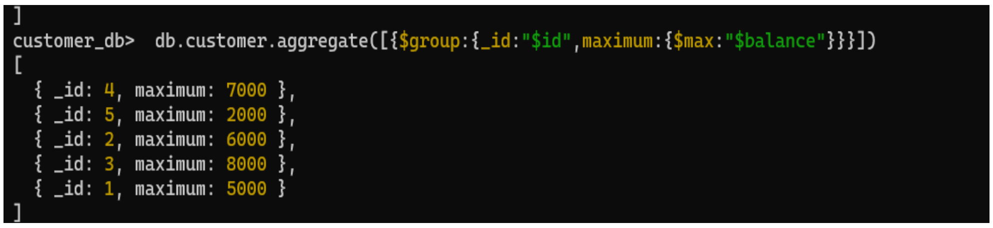
```
[ { _id: null, maximum: 8000 } ]
customer_db>  db.customer.aggregate([{$group:{_id:"$id",average:{$avg:"$balance"}}}])
[
  { _id: 1, average: 5000 },
  { _id: 2, average: 6000 },
  { _id: 5, average: 2000 },
  { _id: 4, average: 7000 },
  { _id: 3, average: 8000 }
]
customer_db>  db.customer.aggregate([{$group:{_id:"$id",maximum:{$max:"$balance"}}}])
[
  { _id: 4, maximum: 7000 },
  { _id: 5, maximum: 2000 },
  { _id: 2, maximum: 6000 },
  { _id: 3, maximum: 8000 },
  { _id: 1, maximum: 5000 }
]
customer_db>  db.customer.aggregate([{$group:{_id:"$sid",minimum:{$min:"$balance"}}}])
[ { _id: null, minimum: 2000 } ]
customer_db>  db.customer.aggregate([{$group:{_id:"$id",minimum:{$min:"$balance"}}}])
[
  { _id: 2, minimum: 6000 },
  { _id: 1, minimum: 5000 },
  { _id: 5, minimum: 2000 },
  { _id: 4, minimum: 7000 },
  { _id: 3, minimum: 8000 }
]
customer_db>
```

Experiment No :4



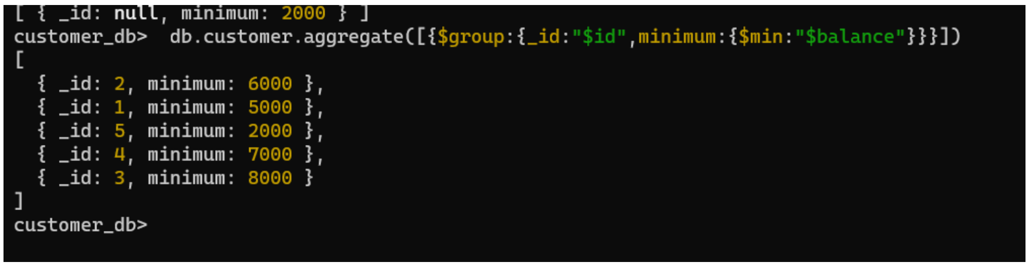
```
customer_db>  db.customer.aggregate([{$group:{_id:"$id",average:{$avg:"$balance"}}}])
[
  { _id: 1, average: 5000 },
  { _id: 2, average: 6000 },
  { _id: 5, average: 2000 },
  { _id: 4, average: 7000 },
  { _id: 3, average: 8000 }
]
```

Q6. compute the maximum of account balance by grouping on customer id for each group.



```
]
customer_db>  db.customer.aggregate([{$group:{_id:"$id",maximum:{$max:"$balance"}}}])
[
  { _id: 4, maximum: 7000 },
  { _id: 5, maximum: 2000 },
  { _id: 2, maximum: 6000 },
  { _id: 3, maximum: 8000 },
  { _id: 1, maximum: 5000 }
]
```

Q7. compute the minimum of account balance by grouping on customer id for each group.

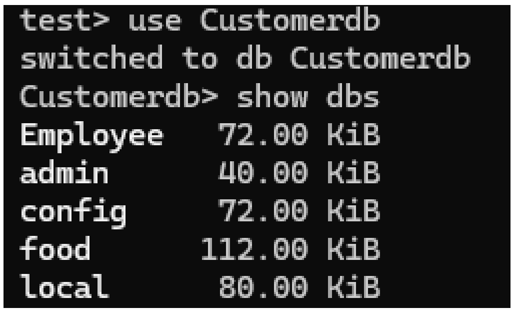


```
[ { _id: null, minimum: 2000 } ]
customer_db>  db.customer.aggregate([{$group:{_id:"$id",minimum:{$min:"$balance"}}}])
[
  { _id: 2, minimum: 6000 },
  { _id: 1, minimum: 5000 },
  { _id: 5, minimum: 2000 },
  { _id: 4, minimum: 7000 },
  { _id: 3, minimum: 8000 }
]
customer_db>
```

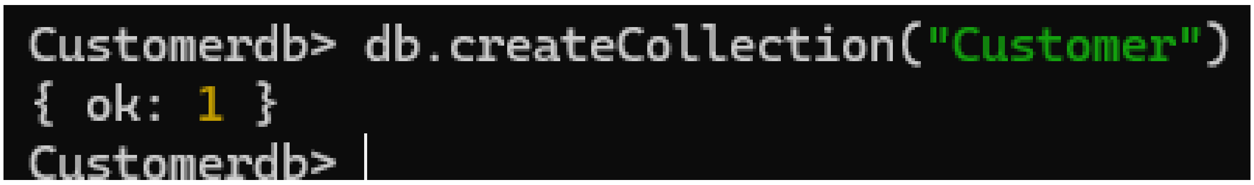
Experiment no: 4

B. Implement Aggregate Function for given set of Question

1. Create a database CustomerdB .Create collection Customer in CustomerdB & print the collection.

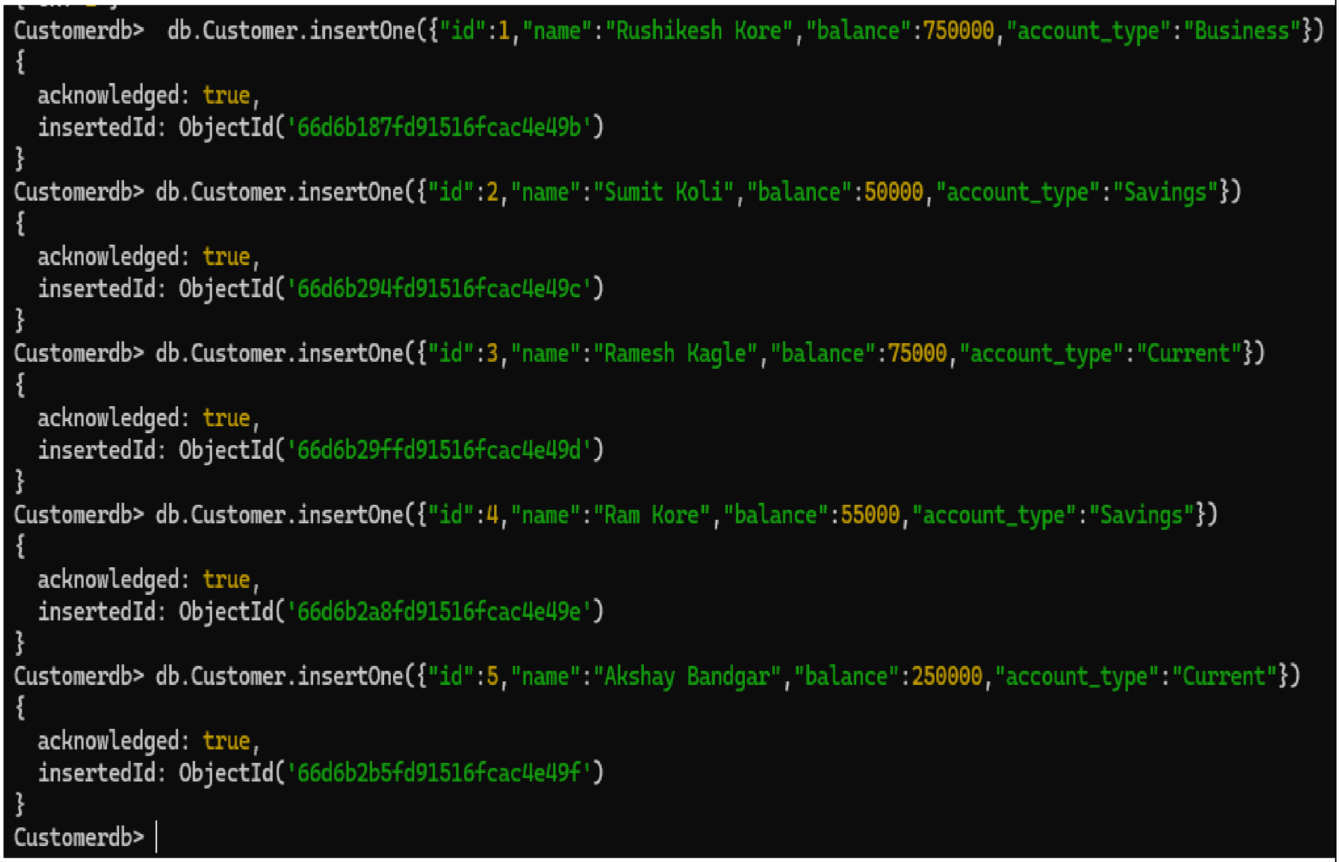


```
test> use Customerdb
switched to db Customerdb
Customerdb> show dbs
Employee    72.00 KiB
admin       40.00 KiB
config      72.00 KiB
food       112.00 KiB
local       80.00 KiB
```



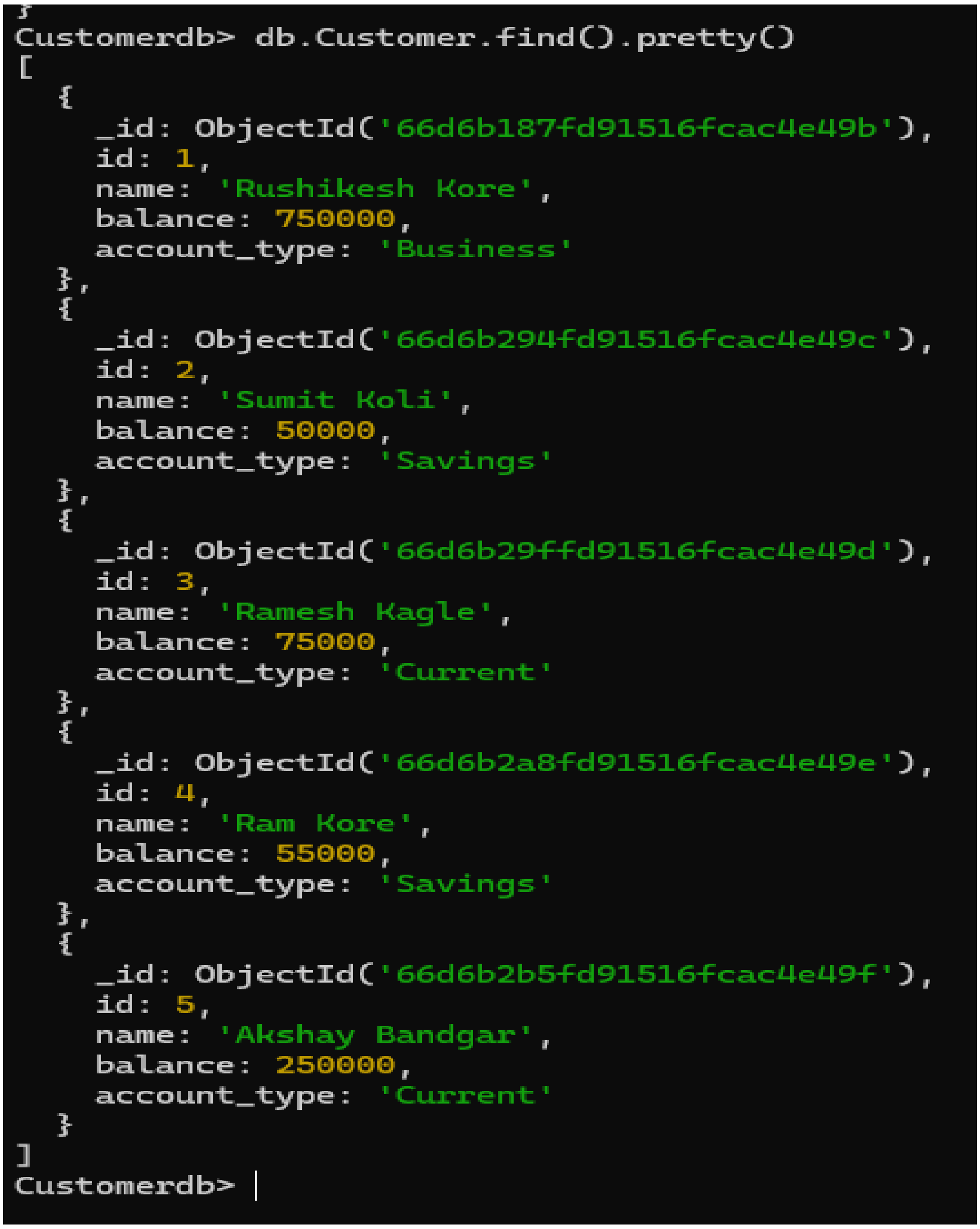
```
Customerdb> db.createCollection("Customer")
{ ok: 1 }
Customerdb>
```

2. Insert following documents in collection Customer (id, name, balance, account type)



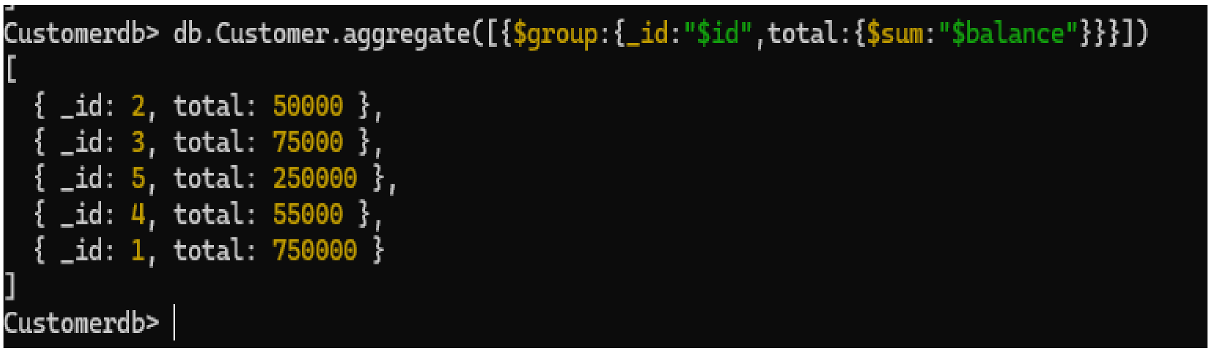
```
Customerdb>  db.Customer.insertOne({"id":1,"name":"Rushikesh Kore","balance":750000,"account_type":"Business"})
{
  acknowledged: true,
  insertedId: ObjectId('66d6b187fd91516fcac4e49b')
}
Customerdb> db.Customer.insertOne({"id":2,"name":"Sumit Koli","balance":50000,"account_type":"Savings"})
{
  acknowledged: true,
  insertedId: ObjectId('66d6b294fd91516fcac4e49c')
}
Customerdb> db.Customer.insertOne({"id":3,"name":"Ramesh Kagle","balance":75000,"account_type":"Current"})
{
  acknowledged: true,
  insertedId: ObjectId('66d6b29ffd91516fcac4e49d')
}
Customerdb> db.Customer.insertOne({"id":4,"name":"Ram Kore","balance":55000,"account_type":"Savings"})
{
  acknowledged: true,
  insertedId: ObjectId('66d6b2a8fd91516fcac4e49e')
}
Customerdb> db.Customer.insertOne({"id":5,"name":"Akshay Bandgar","balance":250000,"account_type":"Current"})
{
  acknowledged: true,
  insertedId: ObjectId('66d6b2b5fd91516fcac4e49f')
}
Customerdb>
```

Print the collection



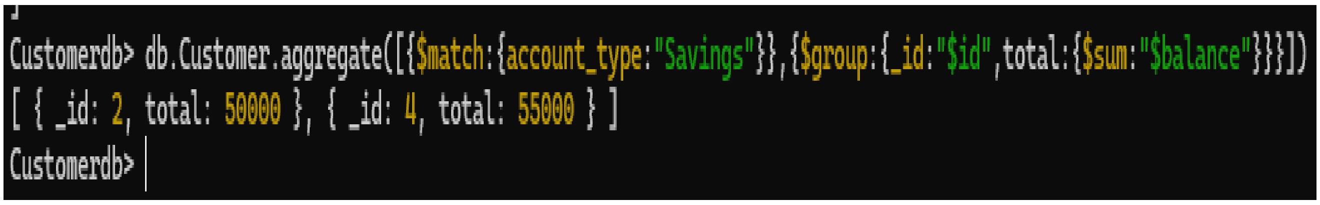
```
Customerdb> db.Customer.find().pretty()
[
  {
    _id: ObjectId('66d6b187fd91516fcac4e49b'),
    id: 1,
    name: 'Rushikesh Kore',
    balance: 750000,
    account_type: 'Business'
  },
  {
    _id: ObjectId('66d6b294fd91516fcac4e49c'),
    id: 2,
    name: 'Sumit Koli',
    balance: 50000,
    account_type: 'Savings'
  },
  {
    _id: ObjectId('66d6b29ffd91516fcac4e49d'),
    id: 3,
    name: 'Ramesh Kagle',
    balance: 75000,
    account_type: 'Current'
  },
  {
    _id: ObjectId('66d6b2a8fd91516fcac4e49e'),
    id: 4,
    name: 'Ram Kore',
    balance: 55000,
    account_type: 'Savings'
  },
  {
    _id: ObjectId('66d6b2b5fd91516fcac4e49f'),
    id: 5,
    name: 'Akshay Bandgar',
    balance: 250000,
    account_type: 'Current'
  }
]
Customerdb> |
```

3. Compute the sum of account balance by grouping on Customer id



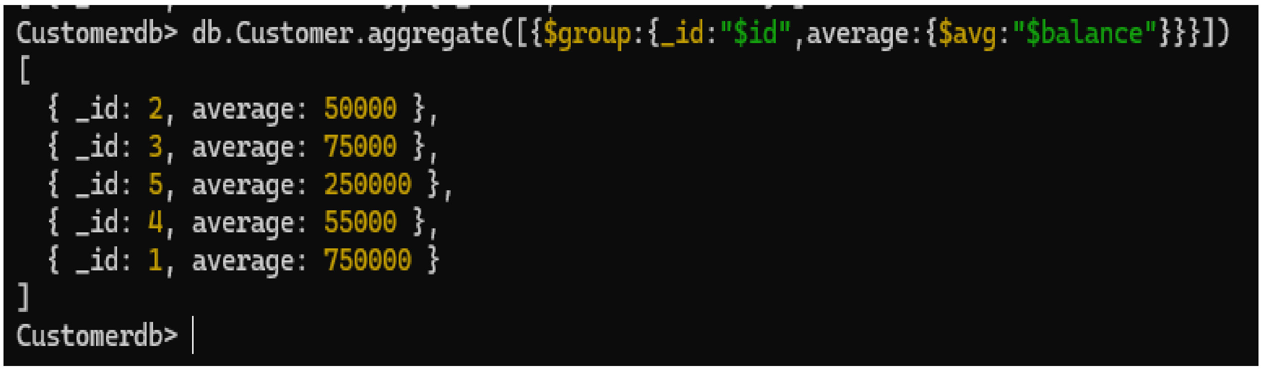
```
Customerdb> db.Customer.aggregate([{$group:{_id:"$id",total:{$sum:"$balance"}}}])
[
  { _id: 2, total: 50000 },
  { _id: 3, total: 75000 },
  { _id: 5, total: 250000 },
  { _id: 4, total: 55000 },
  { _id: 1, total: 750000 }
]
Customerdb> |
```

4. Compute the same of account balance by grouping of customer id use account type is saving Account



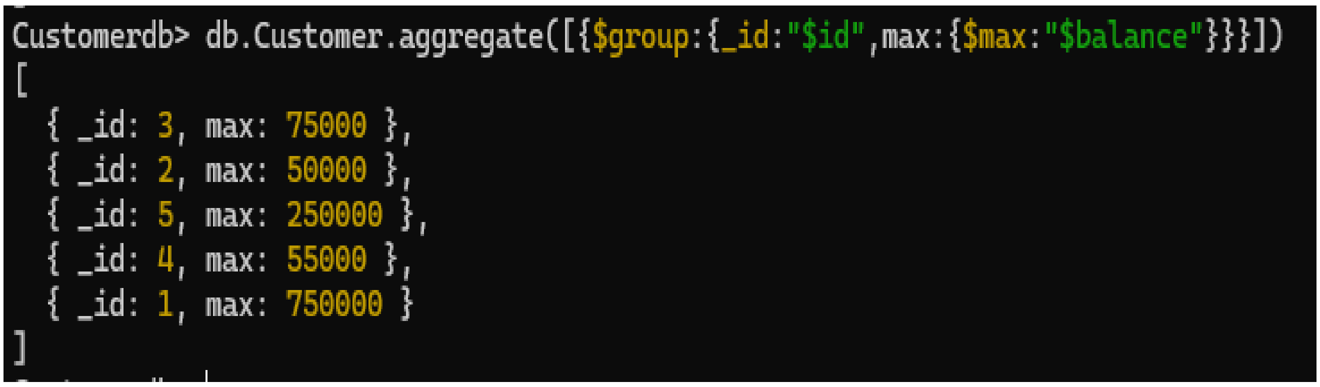
```
Customerdb> db.Customer.aggregate([{$match:{account_type:"Savings"}},{$group:{_id:"$id",total:{$sum:"$balance"}}}])
[ { _id: 2, total: 50000 }, { _id: 4, total: 55000 } ]
Customerdb>
```

5. Compute the Average of account balance by grouping on customer id for each group



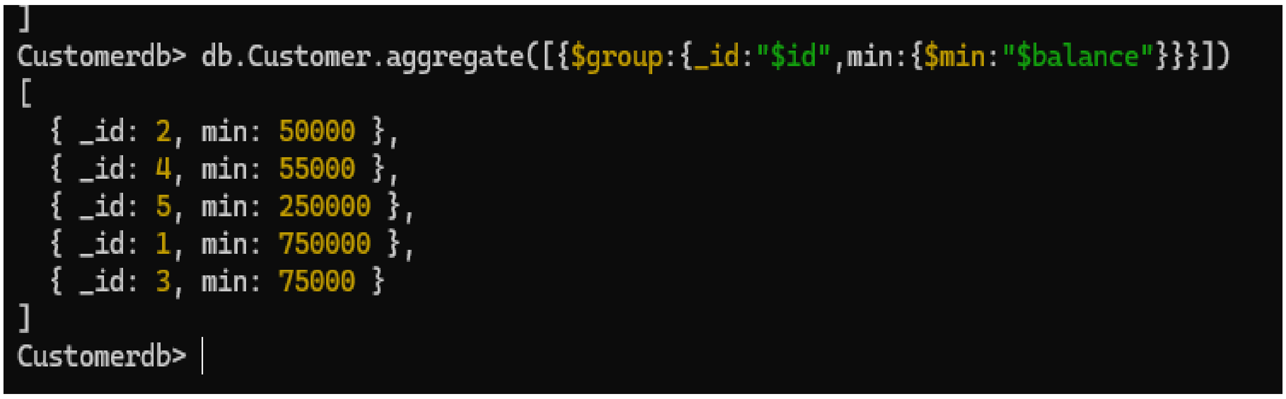
```
Customerdb> db.Customer.aggregate([{$group:{_id:"$id",average:{$avg:"$balance"}}}])
[
  { _id: 2, average: 50000 },
  { _id: 3, average: 75000 },
  { _id: 5, average: 250000 },
  { _id: 4, average: 55000 },
  { _id: 1, average: 750000 }
]
Customerdb>
```

6. Compute the Maximum of account balance by grouping on customer id for each group



```
Customerdb> db.Customer.aggregate([{$group:{_id:"$id",max:{$max:"$balance"}}}])
[
  { _id: 3, max: 75000 },
  { _id: 2, max: 50000 },
  { _id: 5, max: 250000 },
  { _id: 4, max: 55000 },
  { _id: 1, max: 750000 }
]
```

7. Compute the Minimum of account balance by grouping on customer id for each group



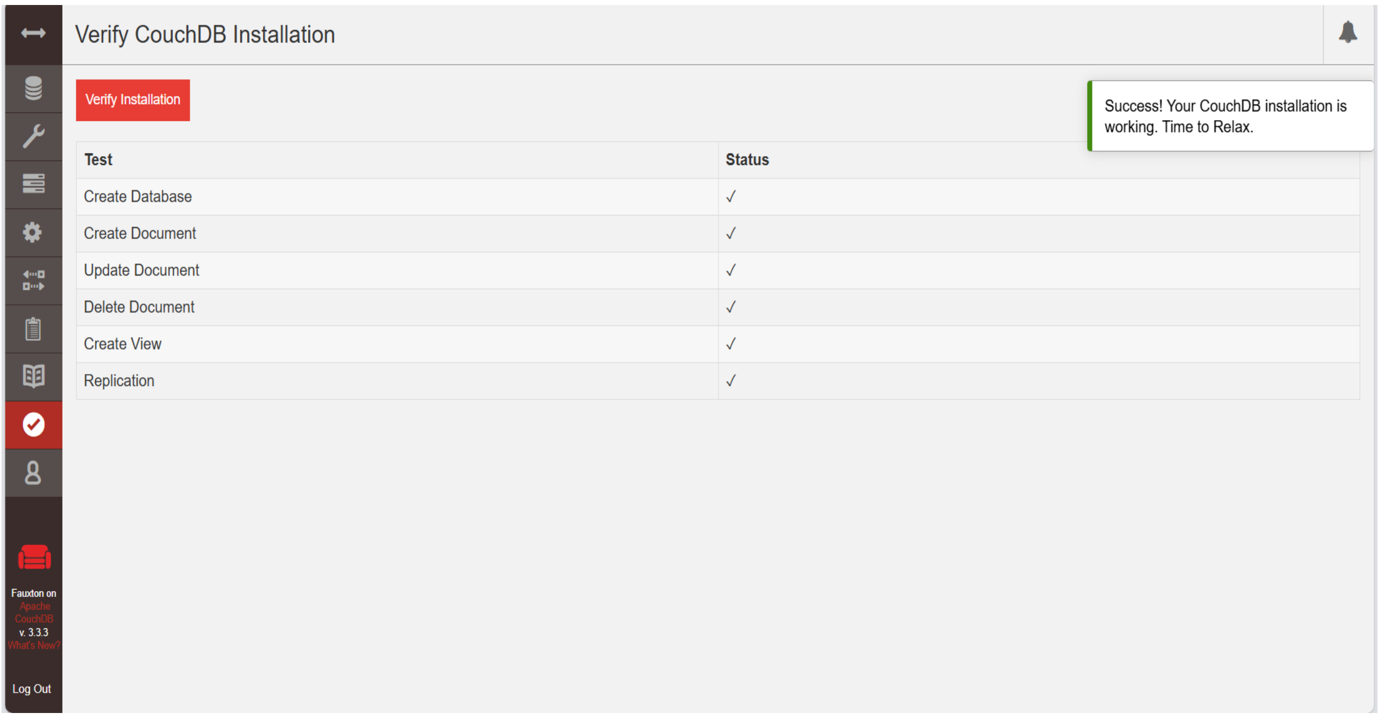
```
]
Customerdb> db.Customer.aggregate([{$group:{_id:"$id",min:{$min:"$balance"}}}])
[
  { _id: 2, min: 50000 },
  { _id: 4, min: 55000 },
  { _id: 5, min: 250000 },
  { _id: 1, min: 750000 },
  { _id: 3, min: 75000 }
]
Customerdb>
```

EXPERIMENT NO: 5

Title: Basic database queries using Apache CouchDB.

Problem statement: Implement CURD operation using CouchDB.

1] Verify CouchDB:



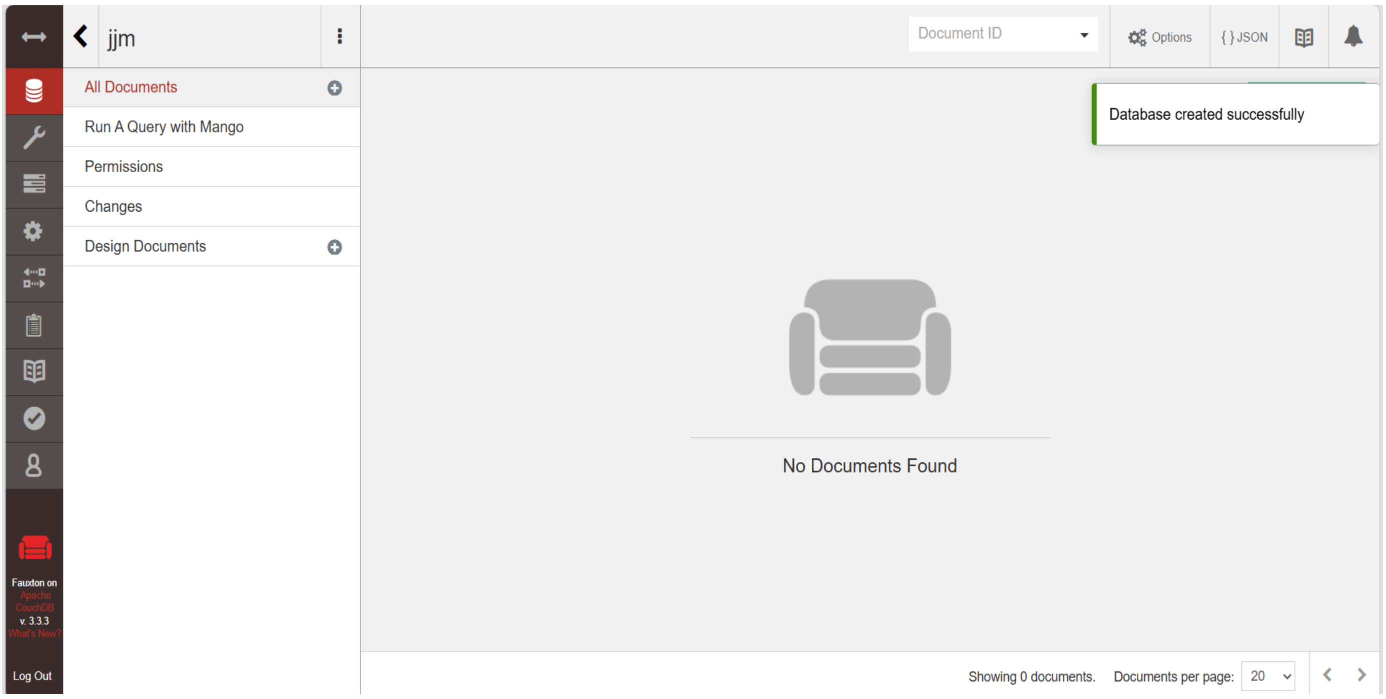
Verify CouchDB Installation

Verify Installation

Success! Your CouchDB installation is working. Time to Relax.

Test	Status
Create Database	√
Create Document	√
Update Document	√
Delete Document	√
Create View	√
Replication	√

2] Create database in CouchDB:



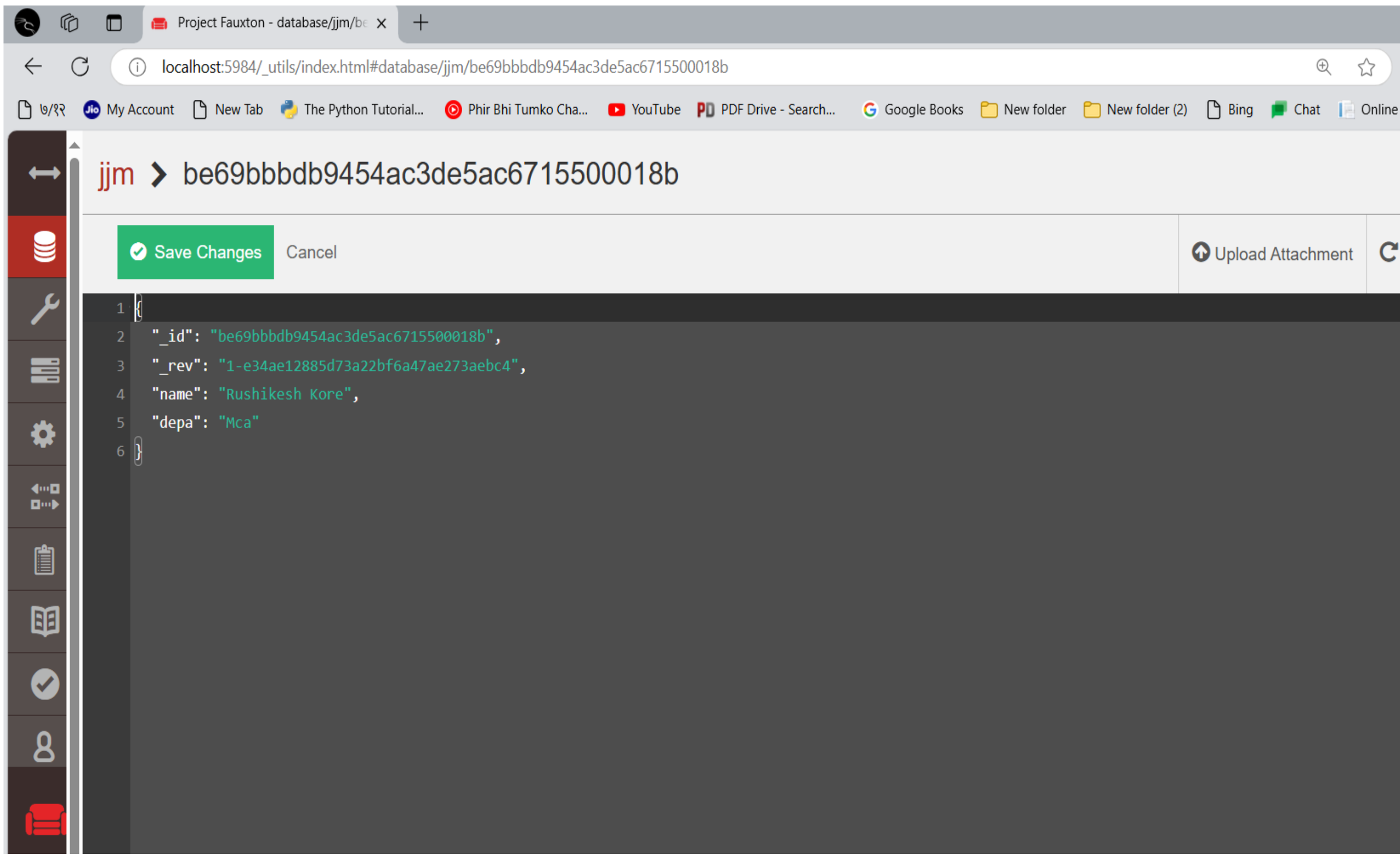
jjm

Database created successfully

No Documents Found

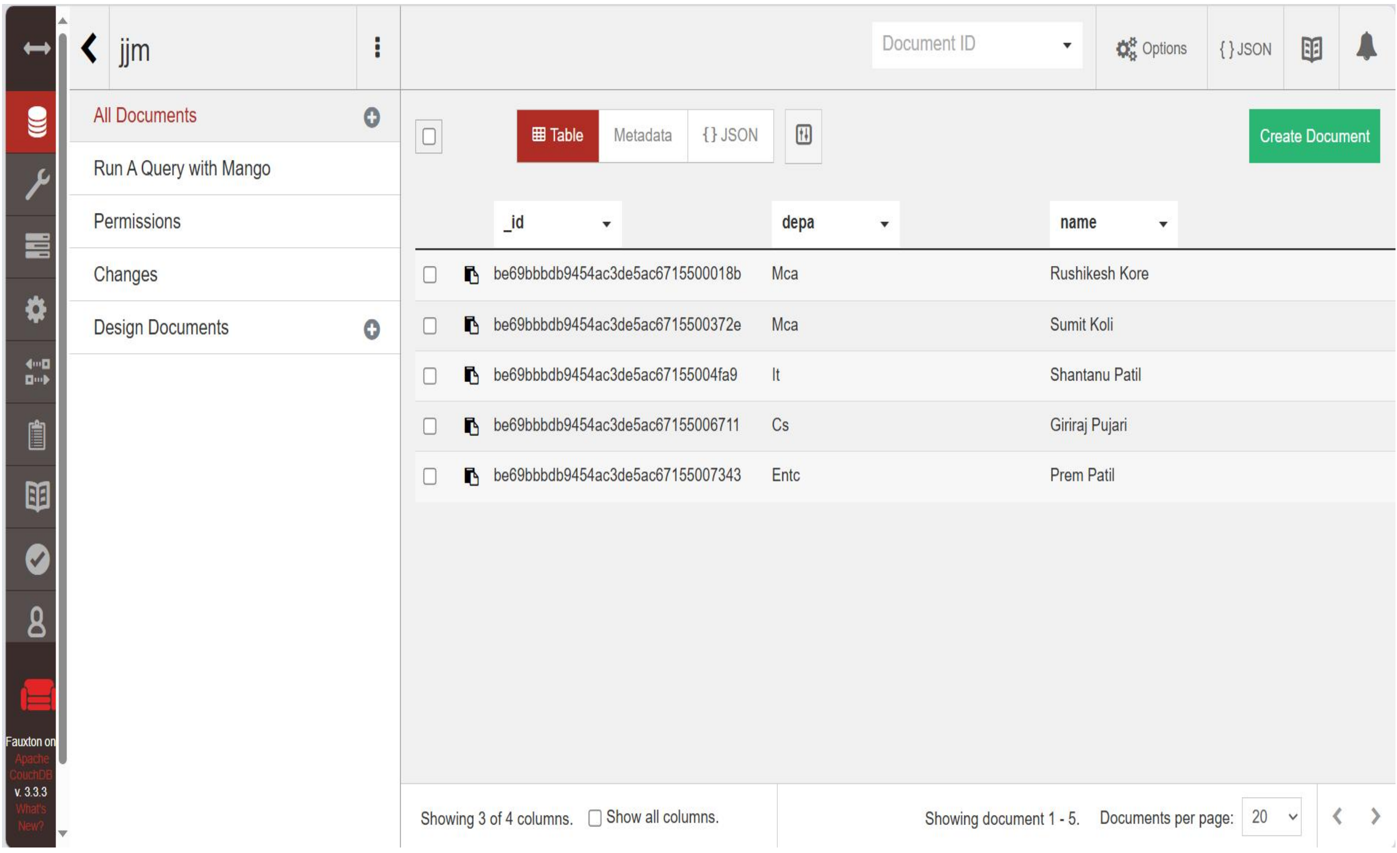
Showing 0 documents. Documents per page: 20

3] Create document in CouchDB:


Project Fauxton - database/jjm/be
localhost:5984/_utils/index.html#database/jjm/be69bbbdb9454ac3de5ac6715500018b
jjm > be69bbbdb9454ac3de5ac6715500018b
Save Changes
Cancel
Upload Attachment

```
{
  "_id": "be69bbbdb9454ac3de5ac6715500018b",
  "_rev": "1-e34ae12885d73a22bf6a47ae273aebc4",
  "name": "Rushikesh Kore",
  "depa": "Mca"
}
```

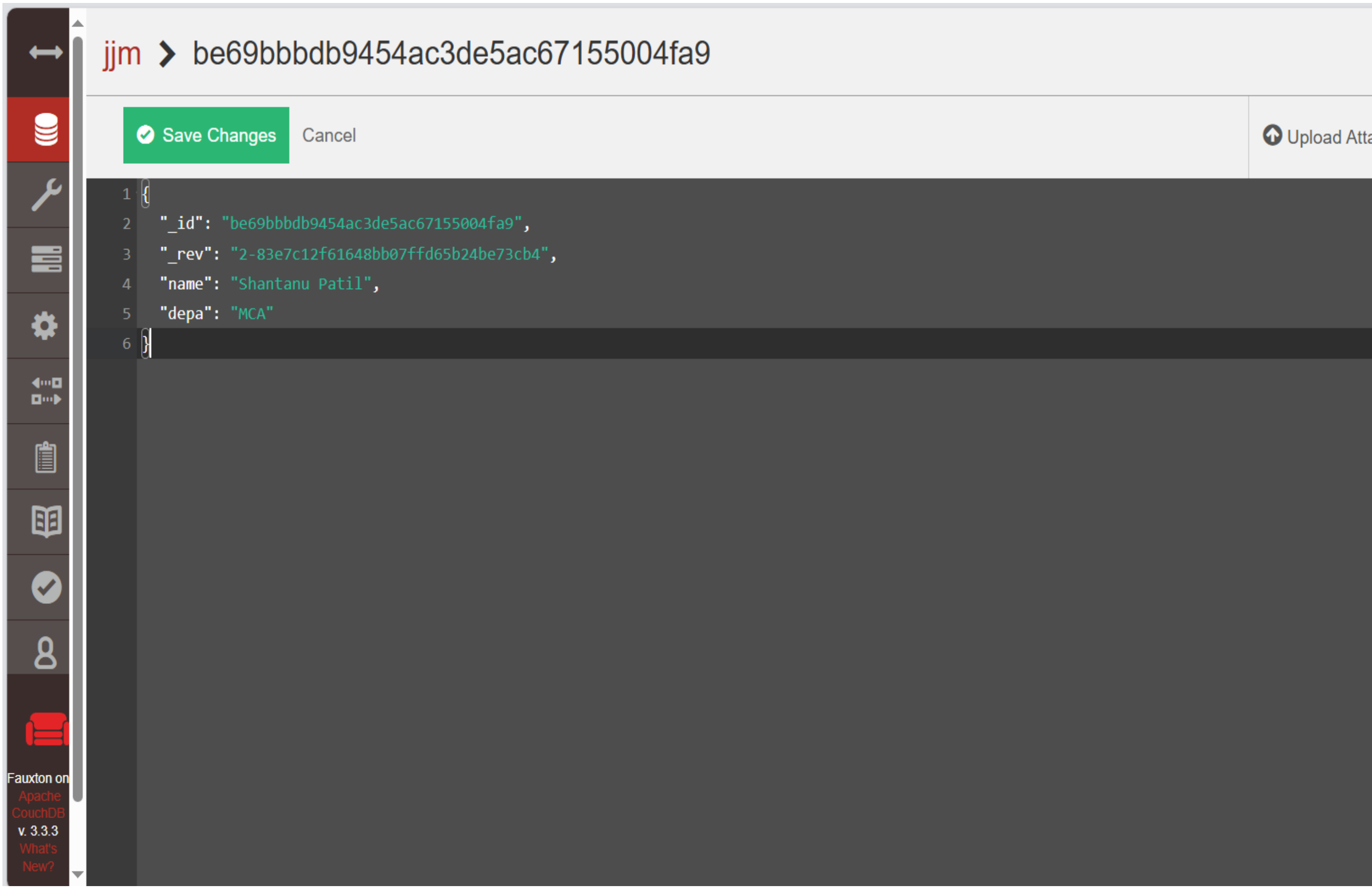
4] Display all document created in CouchDB:



_id	depa	name
be69bbbdb9454ac3de5ac6715500018b	Mca	Rushikesh Kore
be69bbbdb9454ac3de5ac6715500372e	Mca	Sumit Koli
be69bbbdb9454ac3de5ac67155004fa9	It	Shantanu Patil
be69bbbdb9454ac3de5ac67155006711	Cs	Giriraj Pujari
be69bbbdb9454ac3de5ac67155007343	Entc	Prem Patil

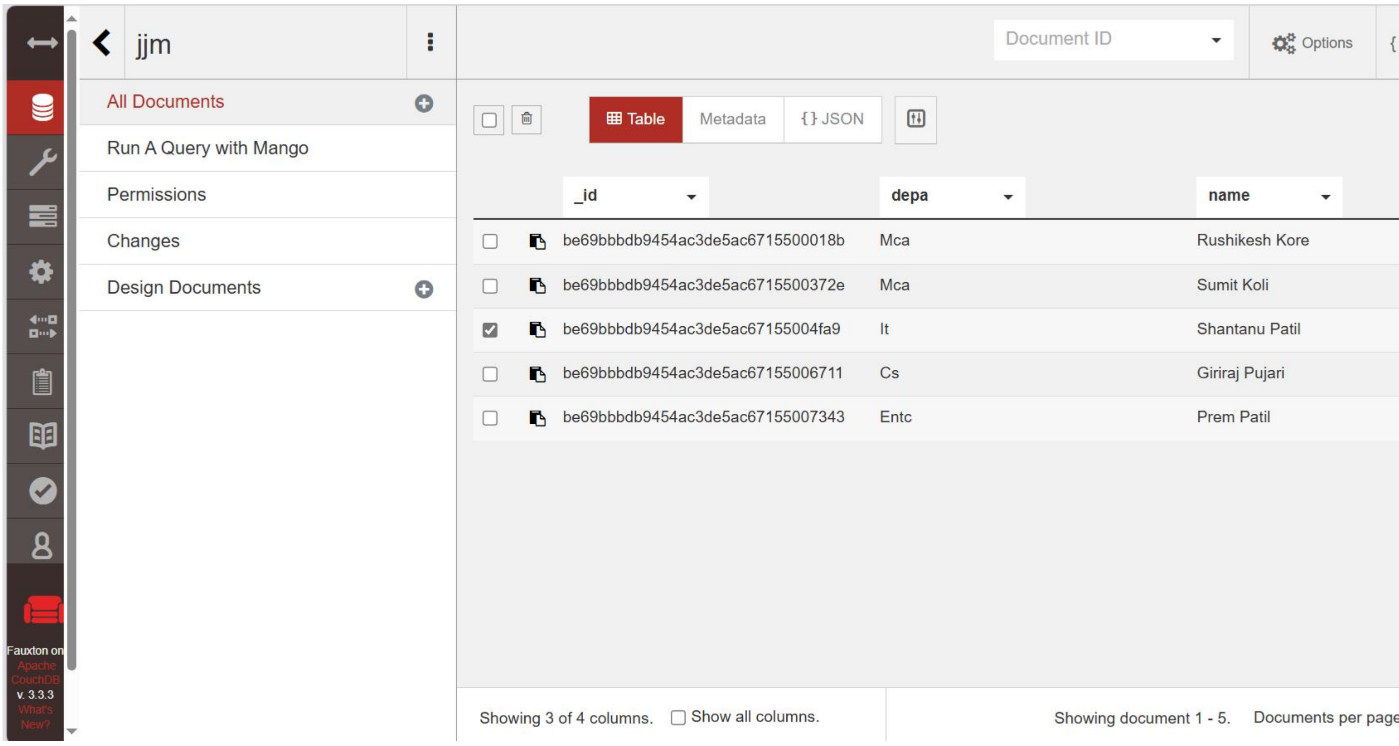
Showing 3 of 4 columns. Show all columns. Showing document 1 - 5. Documents per page: 20

5] Update document in CouchDB:


jjm
be69bbbdb9454ac3de5ac67155004fa9
Save Changes
Cancel
Upload Atta
Fauxton on Apache CouchDB v. 3.3.3 What's New?

```
{
  "_id": "be69bbbdb9454ac3de5ac67155004fa9",
  "_rev": "2-83e7c12f61648bb07ffd65b24be73cb4",
  "name": "Shantanu Patil",
  "depa": "MCA"
}
```

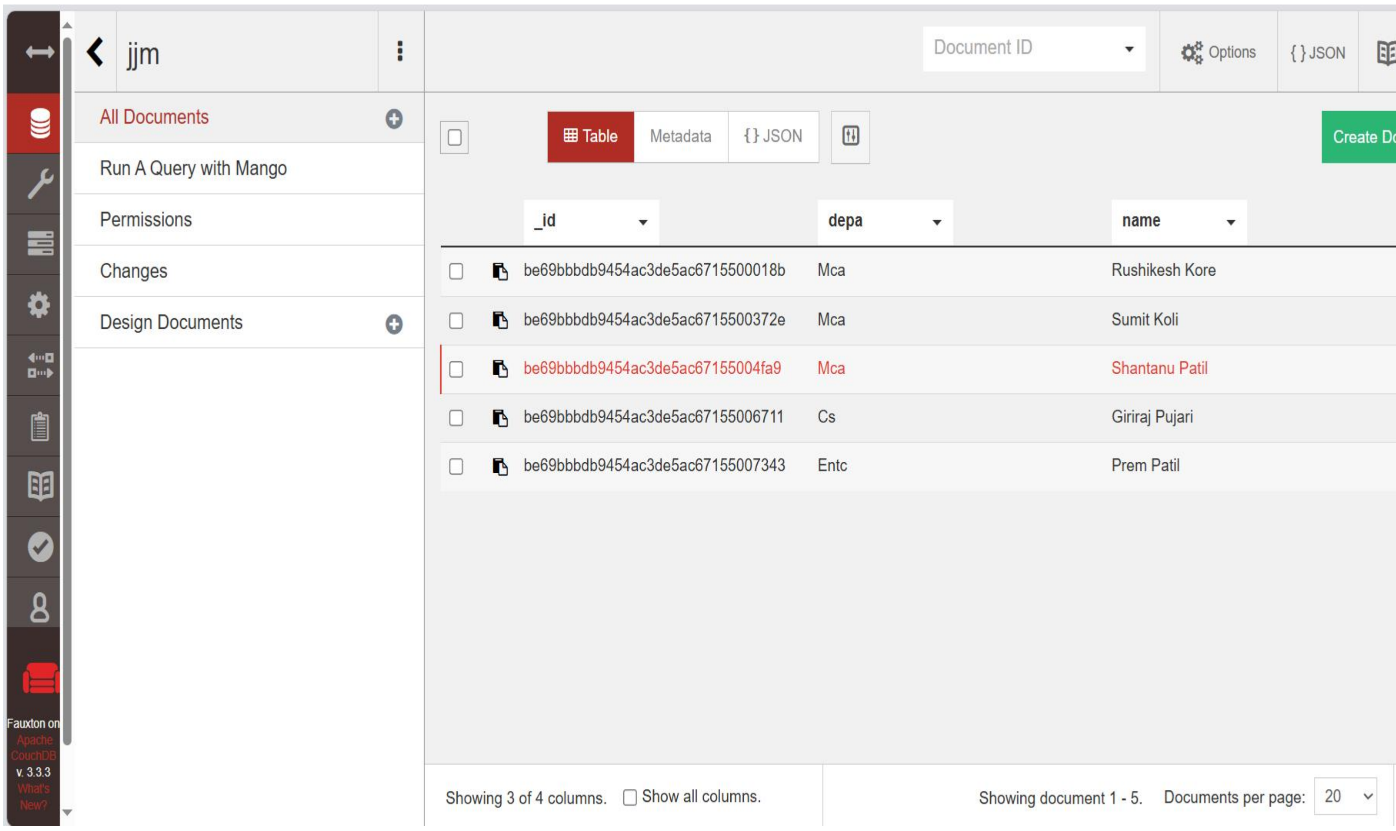
Existing Document:



_id	depa	name
be69bbbdb9454ac3de5ac6715500018b	Mca	Rushikesh Kore
be69bbbdb9454ac3de5ac6715500372e	Mca	Sumit Koli
be69bbbdb9454ac3de5ac67155004fa9	It	Shantanu Patil
be69bbbdb9454ac3de5ac67155006711	Cs	Giriraj Pujari
be69bbbdb9454ac3de5ac67155007343	Entc	Prem Patil

Showing 3 of 4 columns. Show all columns. Showing document 1 - 5. Documents per page

Updated Document:

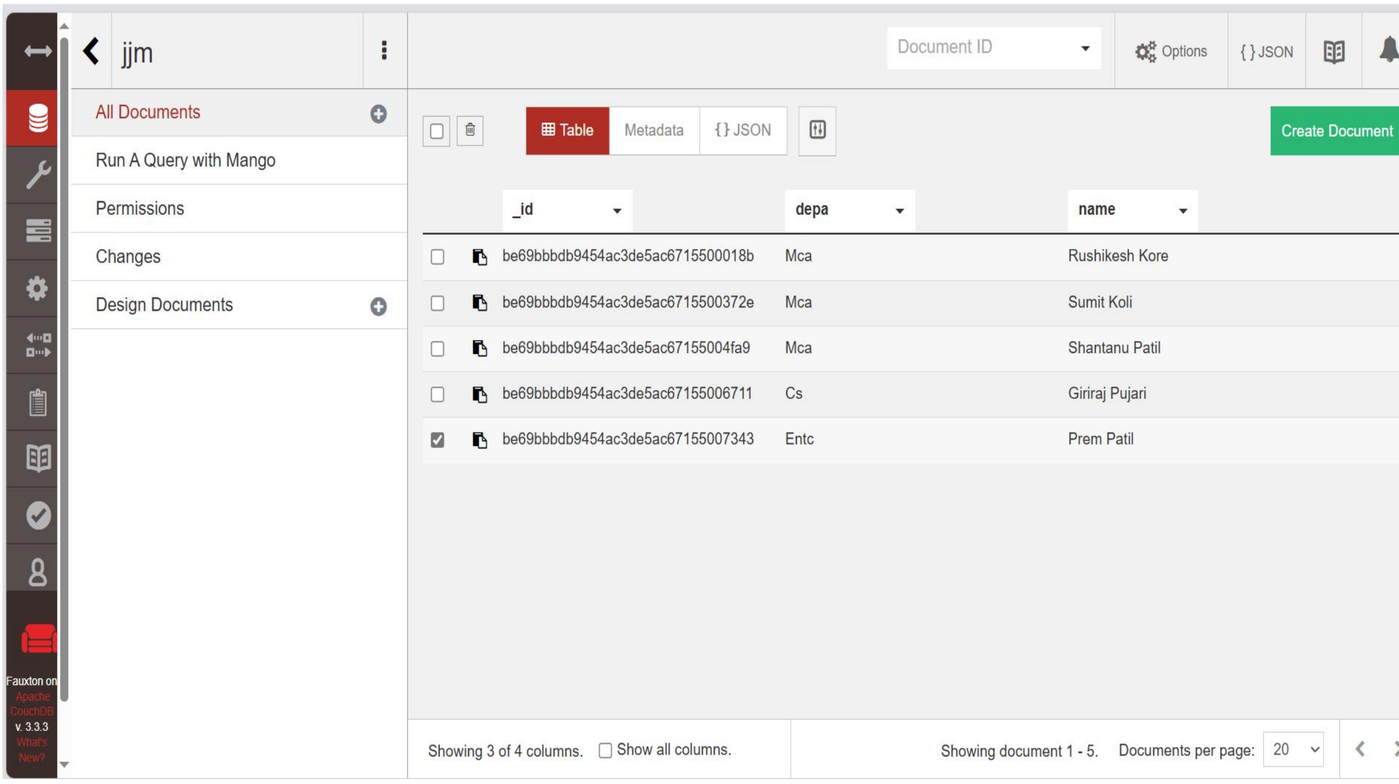


_id	depa	name
be69bbbdb9454ac3de5ac6715500018b	Mca	Rushikesh Kore
be69bbbdb9454ac3de5ac6715500372e	Mca	Sumit Koli
be69bbbdb9454ac3de5ac67155004fa9	Mca	Shantanu Patil
be69bbbdb9454ac3de5ac67155006711	Cs	Giriraj Pujari
be69bbbdb9454ac3de5ac67155007343	Entc	Prem Patil

Showing 3 of 4 columns. Show all columns. Showing document 1 - 5. Documents per page: 20

6] Delete document in CouchDB:

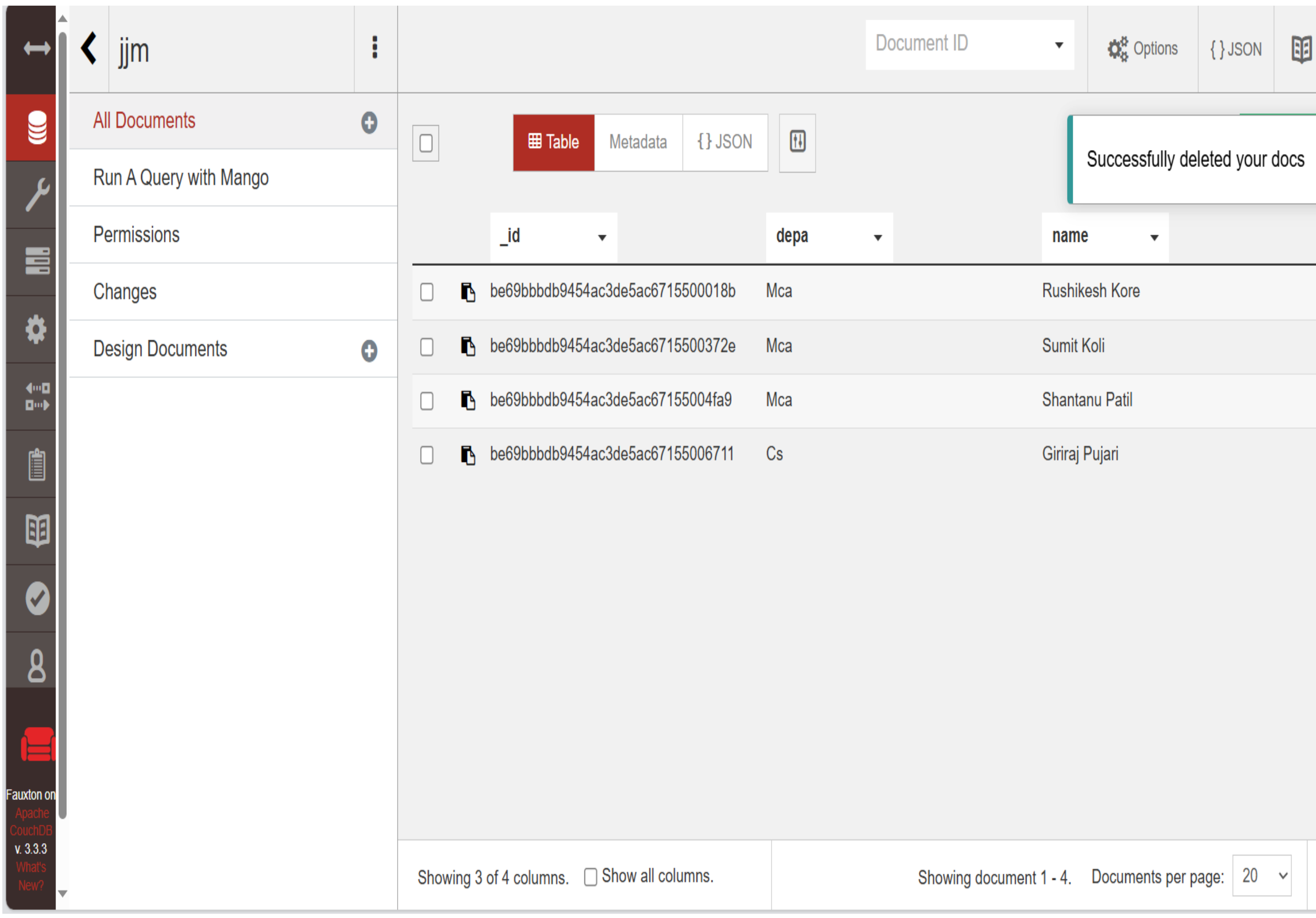
Existing Document:



	_id	depa	name
☐	be69bbbdb9454ac3de5ac6715500018b	Mca	Rushikesh Kore
☐	be69bbbdb9454ac3de5ac6715500372e	Mca	Sumit Koli
☐	be69bbbdb9454ac3de5ac67155004fa9	Mca	Shantanu Patil
☐	be69bbbdb9454ac3de5ac67155006711	Cs	Giriraj Pujari
✓	be69bbbdb9454ac3de5ac67155007343	Entc	Prem Patil

Showing 3 of 4 columns. Show all columns. Showing document 1 - 5. Documents per page: 20

Updated Document:



Successfully deleted your docs

_id	depa	name
be69bbbdb9454ac3de5ac6715500018b	Mca	Rushikesh Kore
be69bbbdb9454ac3de5ac6715500372e	Mca	Sumit Koli
be69bbbdb9454ac3de5ac67155004fa9	Mca	Shantanu Patil
be69bbbdb9454ac3de5ac67155006711	Cs	Giriraj Pujari

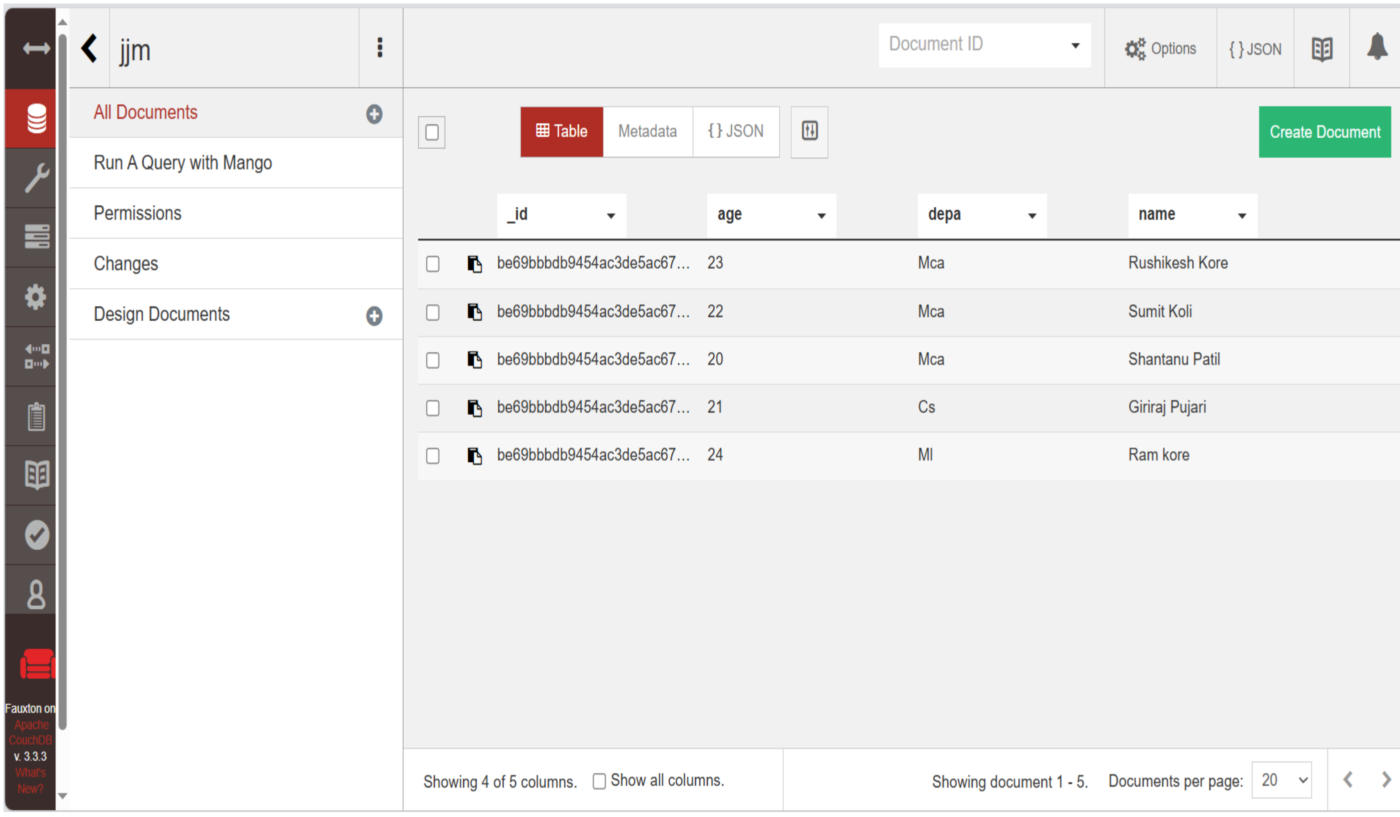
Showing 3 of 4 columns. Show all columns.

Showing document 1 - 4. Documents per page: 20

EXPERIMENT NO: 6

Title: Implement Apache CouchDB view and MapReduce.

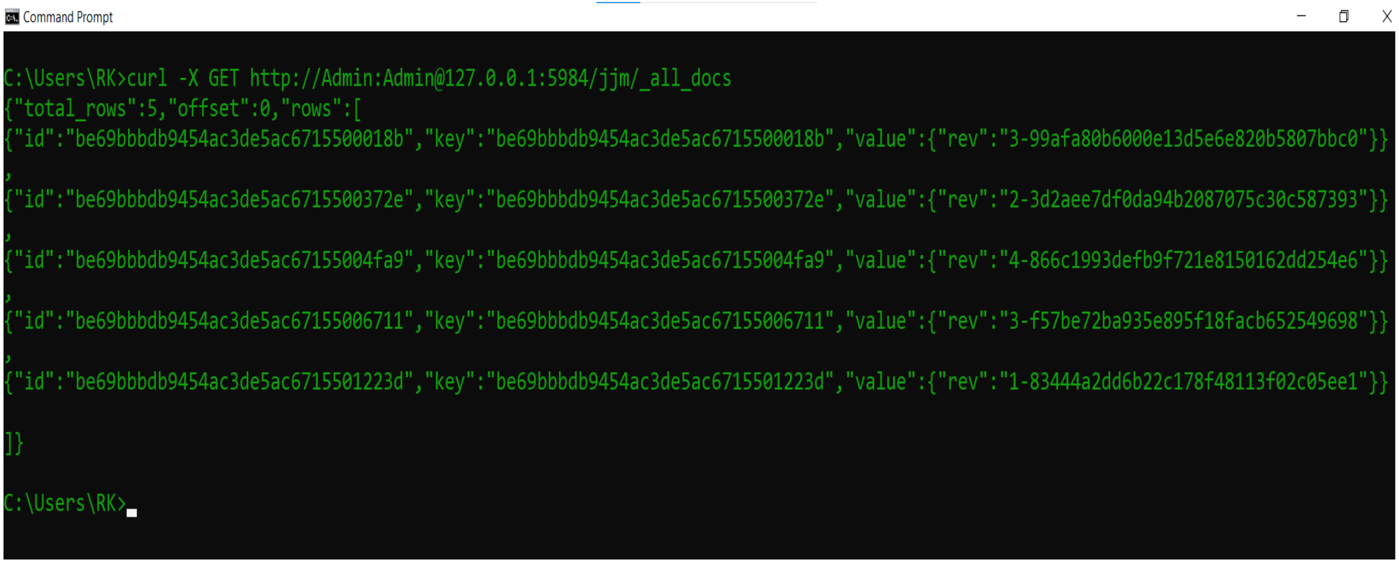
1] Create database in CouchDB.



_id	age	depa	name
be69bbbdb9454ac3de5ac67...	23	Mca	Rushikesh Kore
be69bbbdb9454ac3de5ac67...	22	Mca	Sumit Koli
be69bbbdb9454ac3de5ac67...	20	Mca	Shantanu Patil
be69bbbdb9454ac3de5ac67...	21	Cs	Giriraj Pujari
be69bbbdb9454ac3de5ac67...	24	Ml	Ram kore

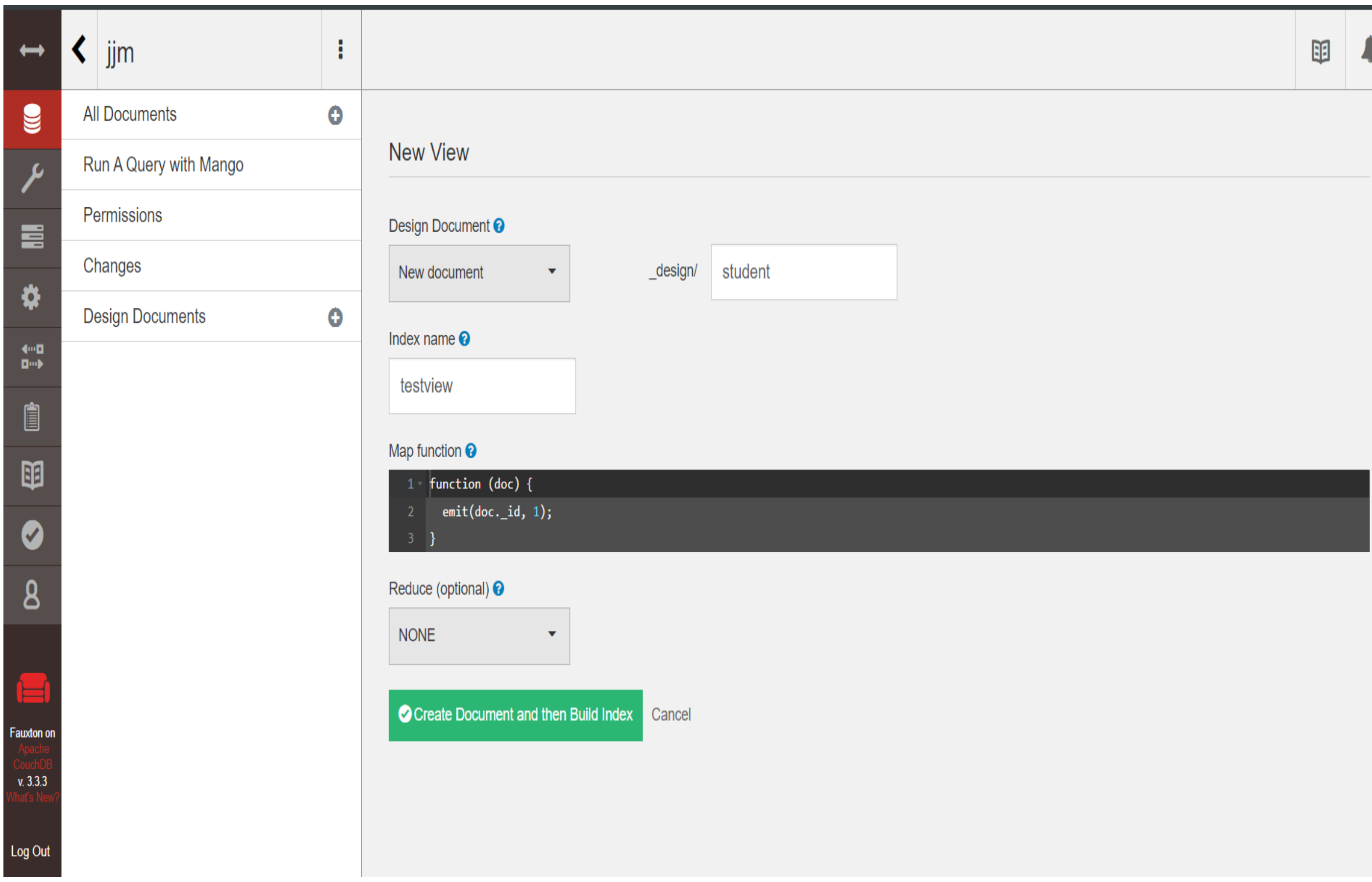
Showing 4 of 5 columns. Show all columns. Showing document 1 - 5. Documents per page: 20

2] Display all documents created in CouchDB.



```
C:\Users\RK>curl -X GET http://Admin:Admin@127.0.0.1:5984/jjm/_all_docs
{"total_rows":5,"offset":0,"rows":[
{"id":"be69bbbdb9454ac3de5ac6715500018b","key":"be69bbbdb9454ac3de5ac6715500018b","value":{"rev":"3-99afa80b6000e13d5e6e820b5807bbc0"}}
,
{"id":"be69bbbdb9454ac3de5ac6715500372e","key":"be69bbbdb9454ac3de5ac6715500372e","value":{"rev":"2-3d2aee7df0da94b2087075c30c587393"}}
,
{"id":"be69bbbdb9454ac3de5ac67155004fa9","key":"be69bbbdb9454ac3de5ac67155004fa9","value":{"rev":"4-866c1993defb9f721e8150162dd254e6"}}
,
{"id":"be69bbbdb9454ac3de5ac67155006711","key":"be69bbbdb9454ac3de5ac67155006711","value":{"rev":"3-f57be72ba935e895f18facb652549698"}}
,
{"id":"be69bbbdb9454ac3de5ac6715501223d","key":"be69bbbdb9454ac3de5ac6715501223d","value":{"rev":"1-83444a2dd6b22c178f48113f02c05ee1"}}

]}

C:\Users\RK>
```

3] Create view in CouchDB.


jjm

All Documents
Run A Query with Mango
Permissions
Changes
Design Documents

New View

Design Document: New document _design/ student

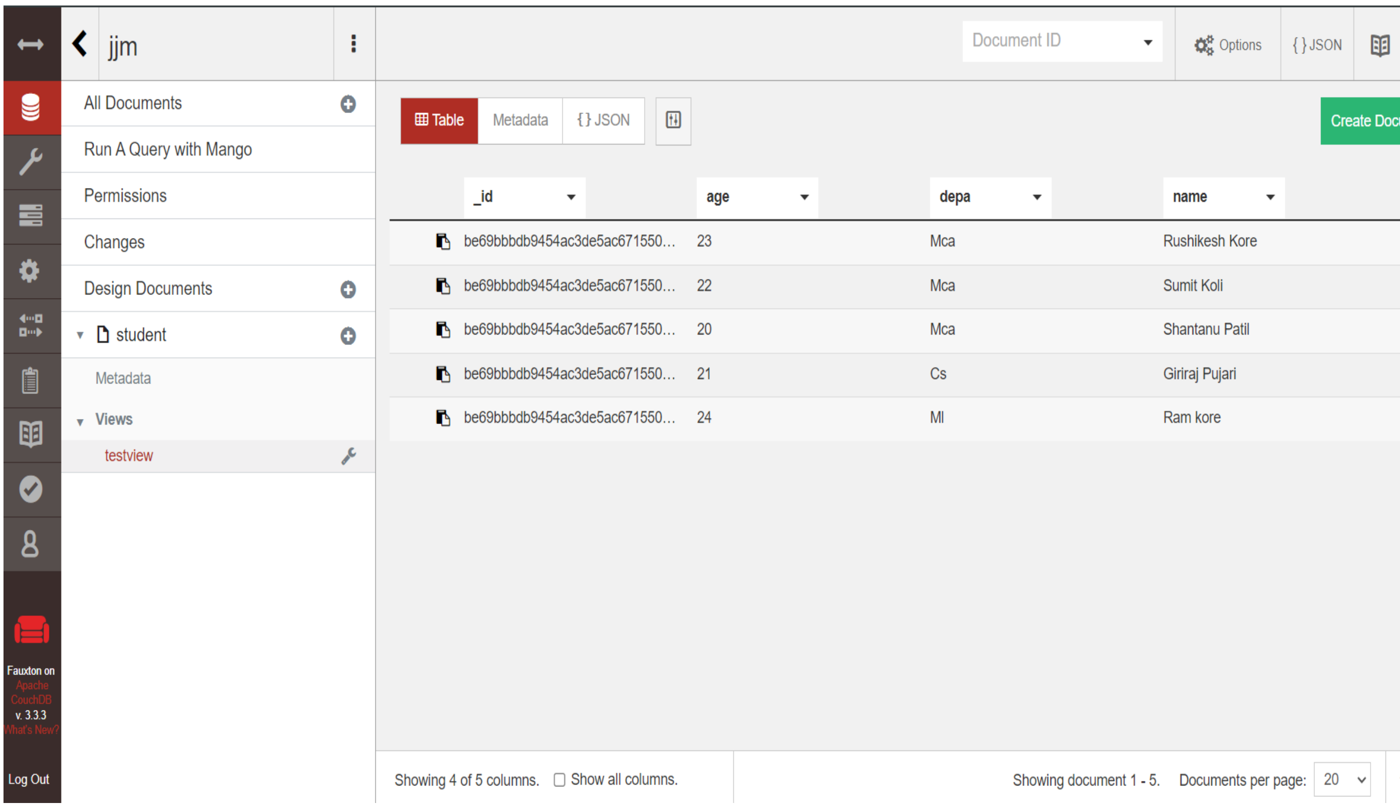
Index name: testview

Map function
```
function (doc) {
  emit(doc._id, 1);
}
```

Reduce (optional): NONE

Create Document and then Build Index Cancel

Fauxton on Apache CouchDB v. 3.3.3 What's New?
Log Out


jjm

All Documents
Run A Query with Mango
Permissions
Changes
Design Documents
student
Metadata
Views
testview

Document ID Options { } JSON

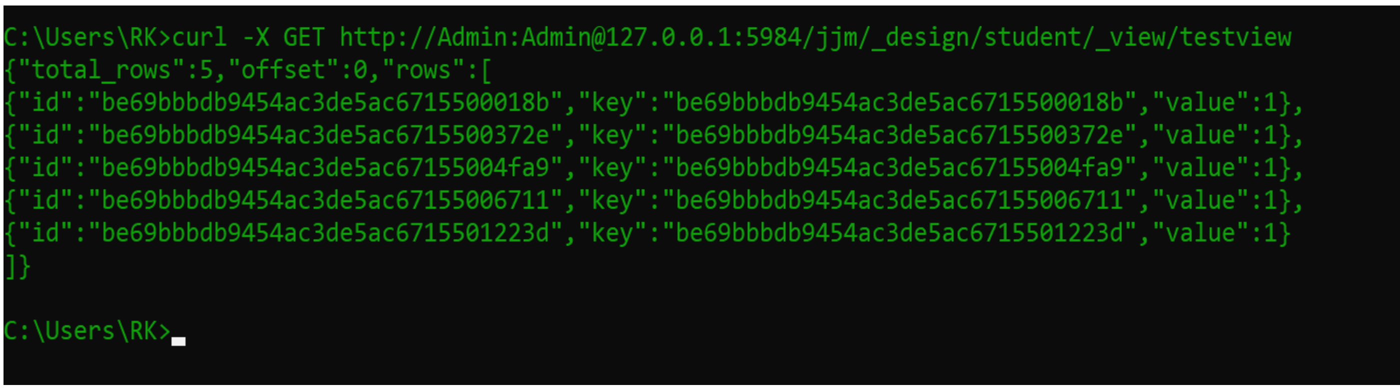
Table Metadata { } JSON Create Doc

_id	age	depa	name
be69bbbdb9454ac3de5ac671550...	23	Mca	Rushikesh Kore
be69bbbdb9454ac3de5ac671550...	22	Mca	Sumit Koli
be69bbbdb9454ac3de5ac671550...	20	Mca	Shantanu Patil
be69bbbdb9454ac3de5ac671550...	21	Cs	Giriraj Pujari
be69bbbdb9454ac3de5ac671550...	24	Ml	Ram kore

Showing 4 of 5 columns. Show all columns. Showing document 1 - 5. Documents per page: 20

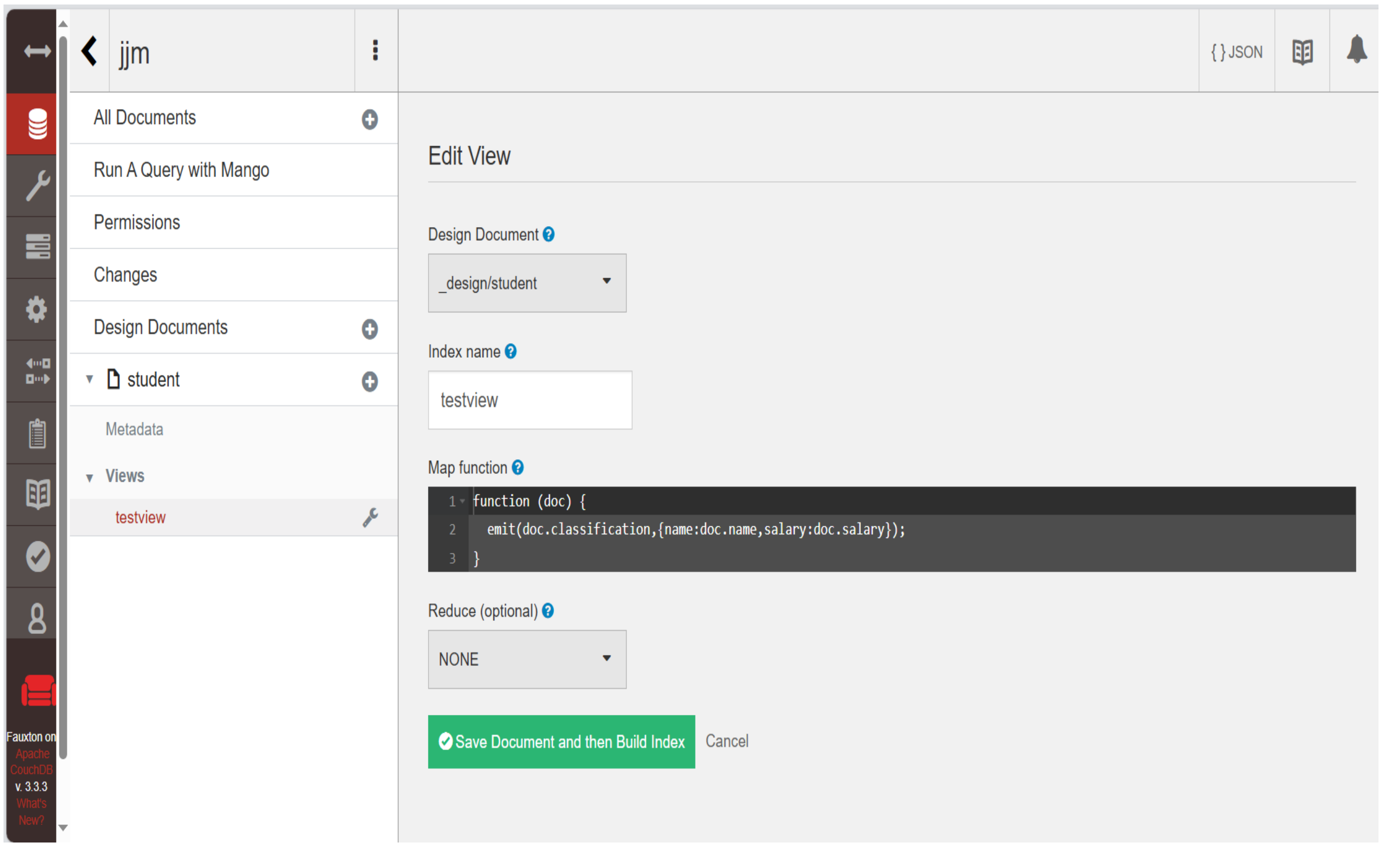
Fauxton on Apache CouchDB v. 3.3.3 What's New?
Log Out

4] Display all documents in view.



```
C:\Users\RK>curl -X GET http://Admin:Admin@127.0.0.1:5984/jjm/_design/student/_view/testview
{"total_rows":5,"offset":0,"rows":[
{"id":"be69bbbdb9454ac3de5ac6715500018b","key":"be69bbbdb9454ac3de5ac6715500018b","value":1},
{"id":"be69bbbdb9454ac3de5ac6715500372e","key":"be69bbbdb9454ac3de5ac6715500372e","value":1},
{"id":"be69bbbdb9454ac3de5ac67155004fa9","key":"be69bbbdb9454ac3de5ac67155004fa9","value":1},
{"id":"be69bbbdb9454ac3de5ac67155006711","key":"be69bbbdb9454ac3de5ac67155006711","value":1},
{"id":"be69bbbdb9454ac3de5ac6715501223d","key":"be69bbbdb9454ac3de5ac6715501223d","value":1}
]}

C:\Users\RK>_
```

5] Edit view in CouchDB.



Edit View

Design Document: _design/student

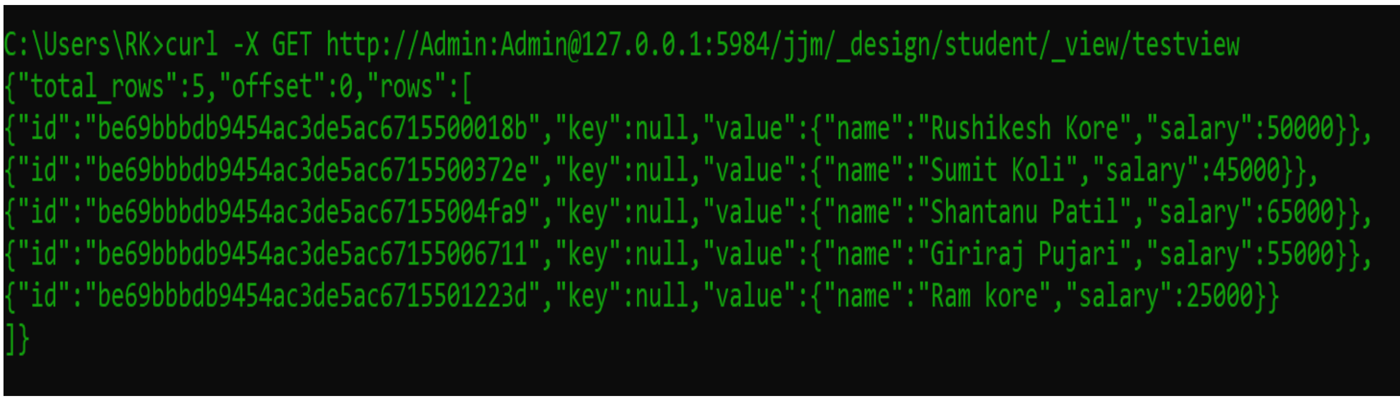
Index name: testview

Map function:

```
function (doc) {
  emit(doc.classification,{name:doc.name,salary:doc.salary});
}
```

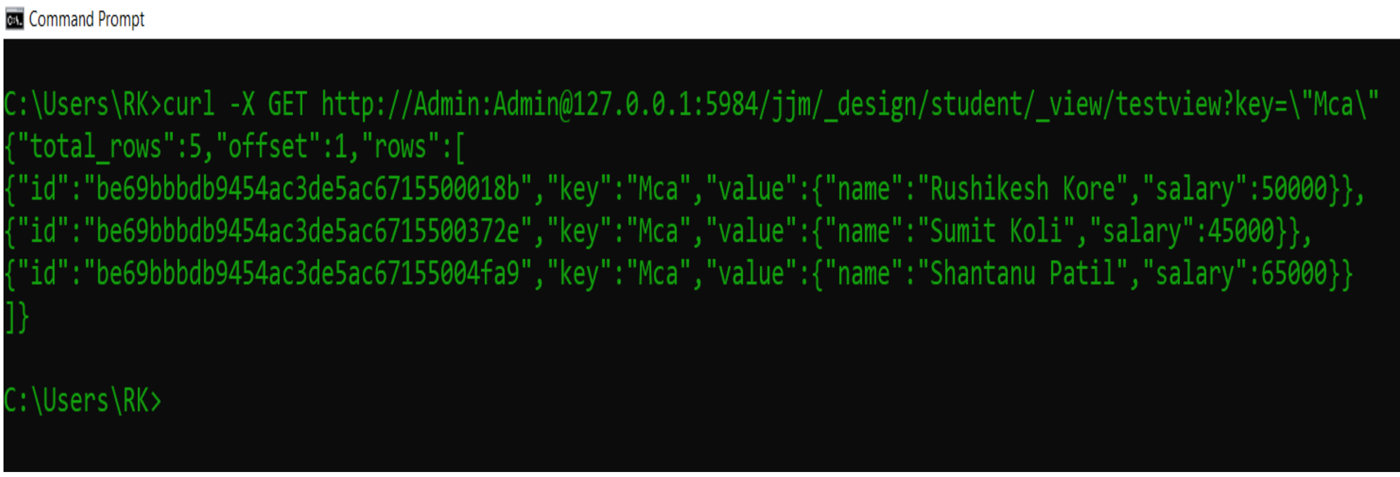
Reduce (optional): NONE

6] Display all documents created in CouchDB.



```
C:\Users\RK>curl -X GET http://Admin:Admin@127.0.0.1:5984/jjm/_design/student/_view/testview
{"total_rows":5,"offset":0,"rows":[
{"id":"be69bbbdb9454ac3de5ac6715500018b","key":null,"value":{"name":"Rushikesh Kore","salary":50000}},
{"id":"be69bbbdb9454ac3de5ac6715500372e","key":null,"value":{"name":"Sumit Koli","salary":45000}},
{"id":"be69bbbdb9454ac3de5ac67155004fa9","key":null,"value":{"name":"Shantanu Patil","salary":65000}},
{"id":"be69bbbdb9454ac3de5ac67155006711","key":null,"value":{"name":"Giriraj Pujari","salary":55000}},
{"id":"be69bbbdb9454ac3de5ac6715501223d","key":null,"value":{"name":"Ram kore","salary":25000}}
]}
```

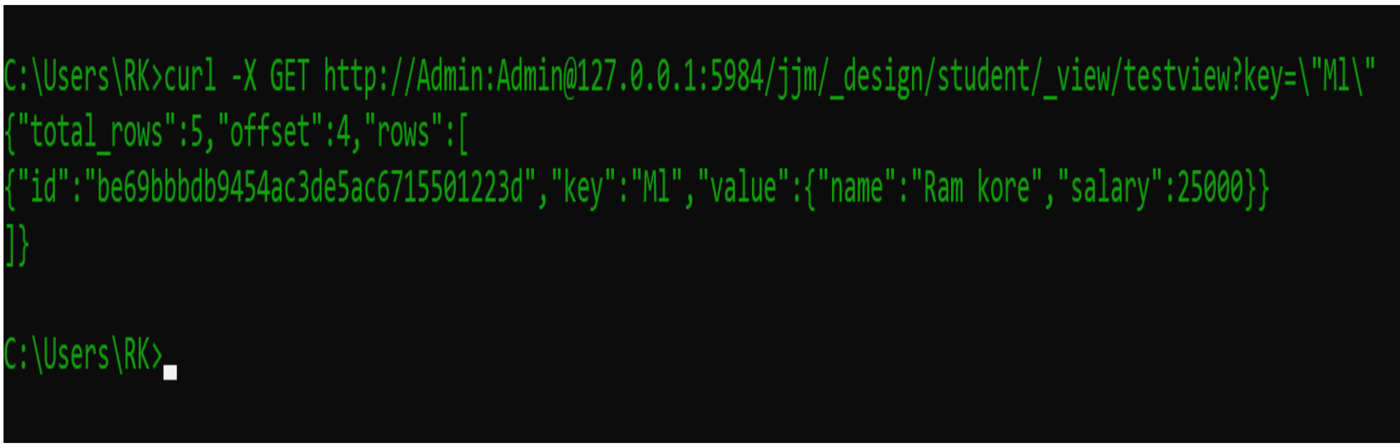
7] Display “Mca” student from documents in CouchDB.



Command Prompt

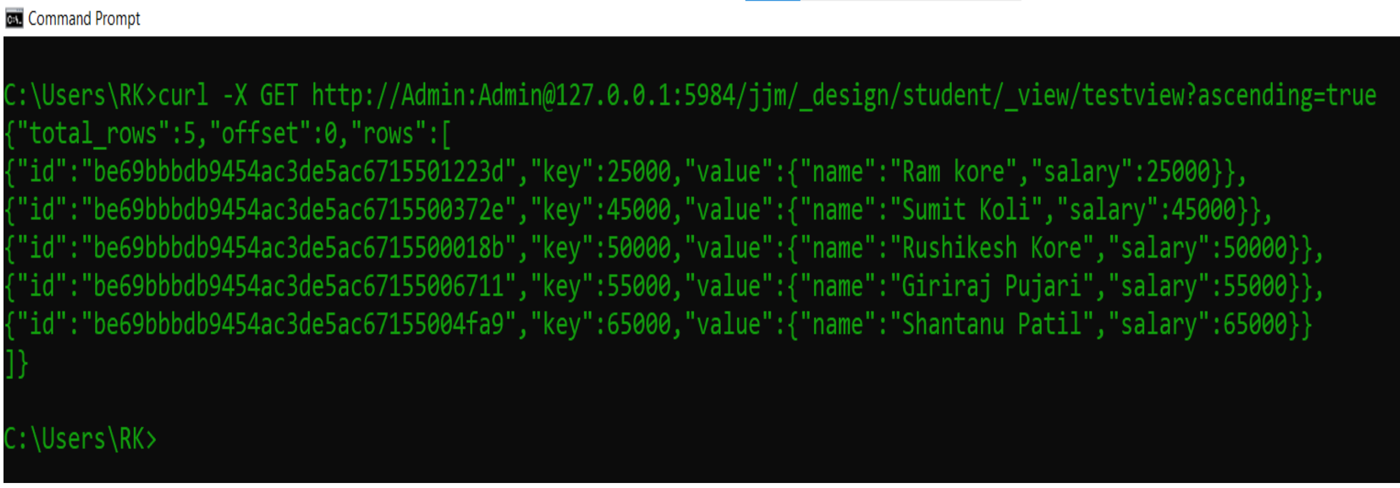
```
C:\Users\RK>curl -X GET http://Admin:Admin@127.0.0.1:5984/jjm/_design/student/_view/testview?key=\"Mca\"
{"total_rows":5,"offset":1,"rows":[
{"id":"be69bbbdb9454ac3de5ac6715500018b","key":"Mca","value":{"name":"Rushikesh Kore","salary":50000}},
{"id":"be69bbbdb9454ac3de5ac6715500372e","key":"Mca","value":{"name":"Sumit Koli","salary":45000}},
{"id":"be69bbbdb9454ac3de5ac67155004fa9","key":"Mca","value":{"name":"Shantanu Patil","salary":65000}}
]}

C:\Users\RK>
```

8] Display “Ml” student from documents in CouchDB.



```
C:\Users\RK>curl -X GET http://Admin:Admin@127.0.0.1:5984/jjm/_design/student/_view/testview?key=\"Ml\"
{"total_rows":5,"offset":4,"rows":[
{"id":"be69bbbdb9454ac3de5ac6715501223d","key":"Ml","value":{"name":"Ram kore","salary":25000}}
]}

C:\Users\RK>
```

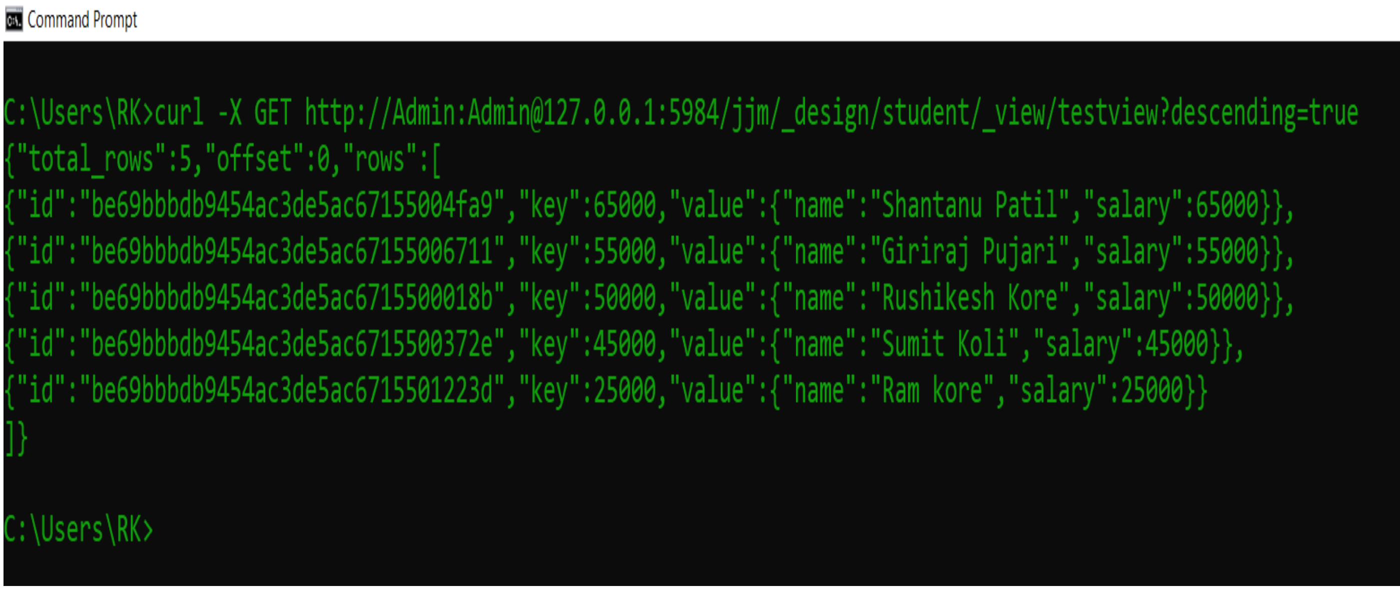
9] Display salary in ascending order.



Command Prompt

```
C:\Users\RK>curl -X GET http://Admin:Admin@127.0.0.1:5984/jjm/_design/student/_view/testview?ascending=true
{"total_rows":5,"offset":0,"rows":[
{"id":"be69bbbdb9454ac3de5ac6715501223d","key":25000,"value":{"name":"Ram kore","salary":25000}},
{"id":"be69bbbdb9454ac3de5ac6715500372e","key":45000,"value":{"name":"Sumit Koli","salary":45000}},
{"id":"be69bbbdb9454ac3de5ac6715500018b","key":50000,"value":{"name":"Rushikesh Kore","salary":50000}},
{"id":"be69bbbdb9454ac3de5ac67155006711","key":55000,"value":{"name":"Giriraj Pujari","salary":55000}},
{"id":"be69bbbdb9454ac3de5ac67155004fa9","key":65000,"value":{"name":"Shantanu Patil","salary":65000}}
]}

C:\Users\RK>
```

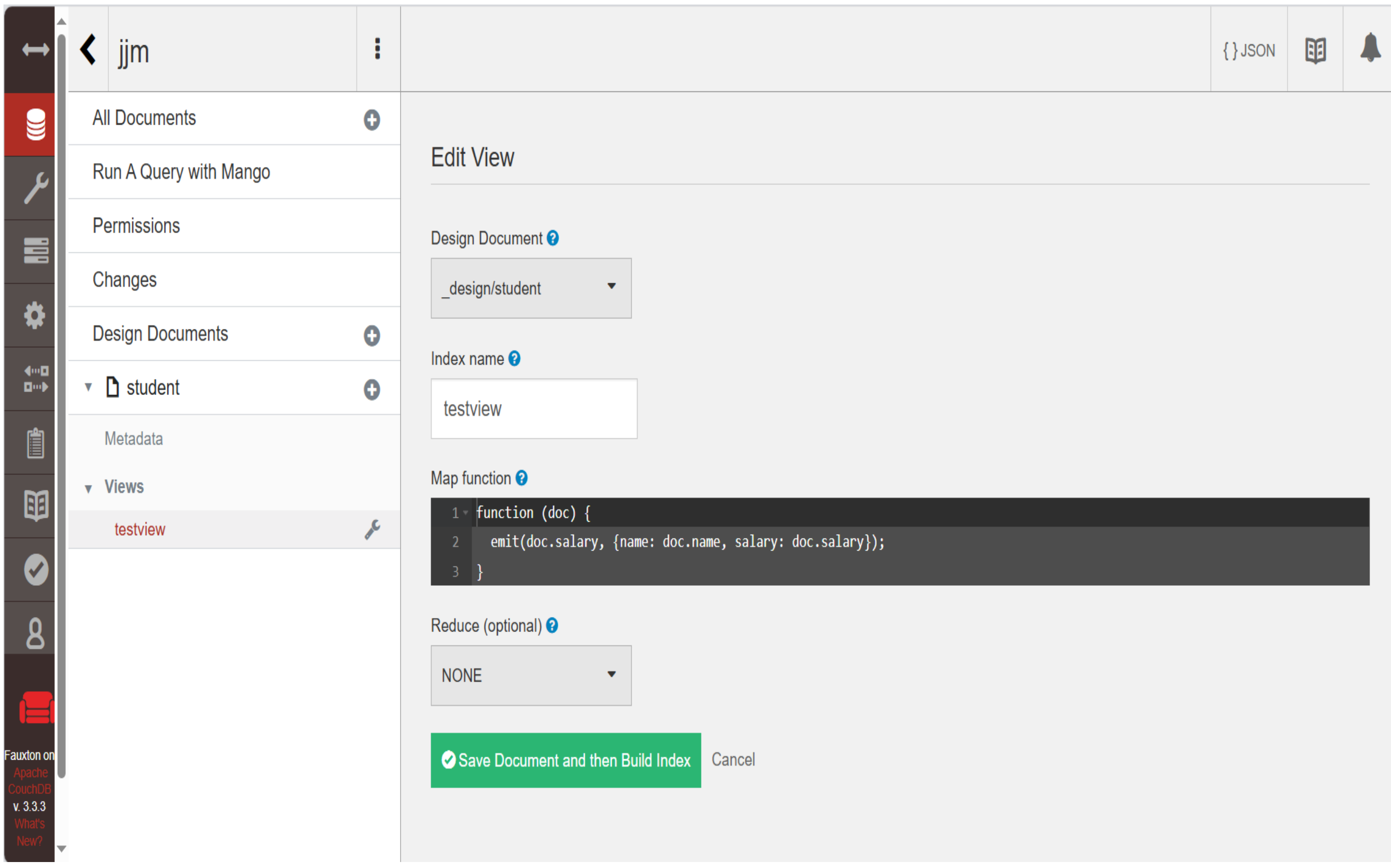
10] Display salary in descending order.



Command Prompt

```
C:\Users\RK>curl -X GET http://Admin:Admin@127.0.0.1:5984/jjm/_design/student/_view/testview?descending=true
{"total_rows":5,"offset":0,"rows":[
{"id":"be69bbbdb9454ac3de5ac67155004fa9","key":65000,"value":{"name":"Shantanu Patil","salary":65000}},
{"id":"be69bbbdb9454ac3de5ac67155006711","key":55000,"value":{"name":"Giriraj Pujari","salary":55000}},
{"id":"be69bbbdb9454ac3de5ac6715500018b","key":50000,"value":{"name":"Rushikesh Kore","salary":50000}},
{"id":"be69bbbdb9454ac3de5ac6715500372e","key":45000,"value":{"name":"Sumit Koli","salary":45000}},
{"id":"be69bbbdb9454ac3de5ac6715501223d","key":25000,"value":{"name":"Ram kore","salary":25000}}
]}

C:\Users\RK>
```

11] Edit view in CouchDB.



Edit View

Design Document: _design/student

Index name: testview

Map function

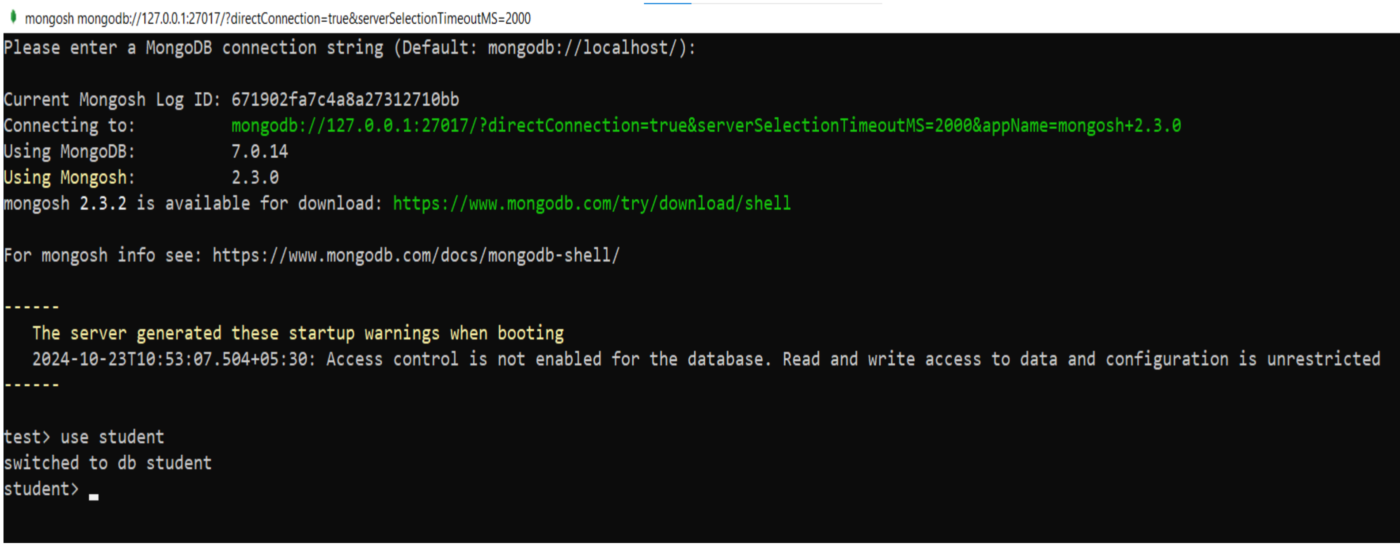
```
function (doc) {
  emit(doc.salary, {name: doc.name, salary: doc.salary});
}
```

Reduce (optional): NONE

EXPERIMENT NO: 7

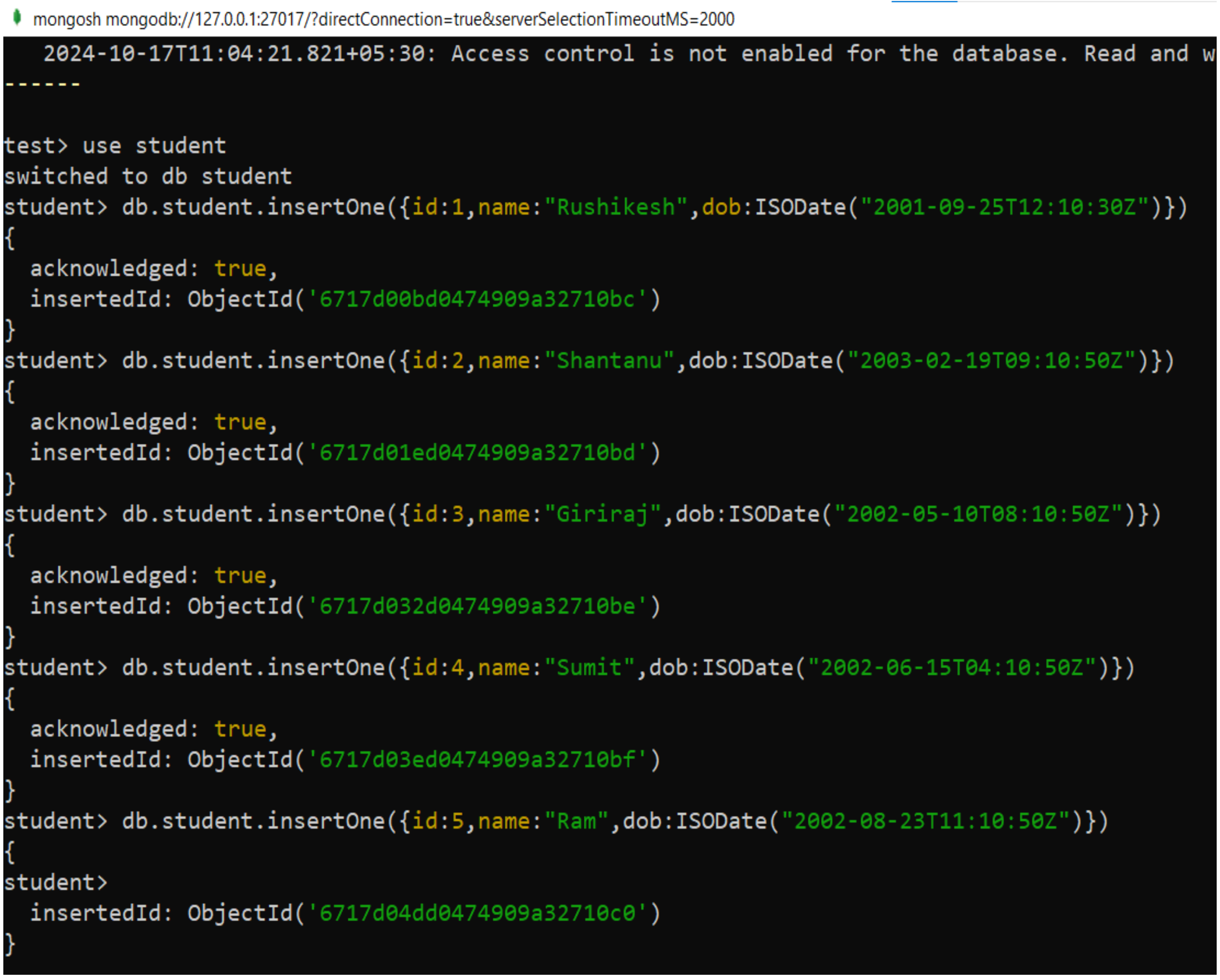
Title: Implement data queries in MongoDB.

Q1. Create a database student.

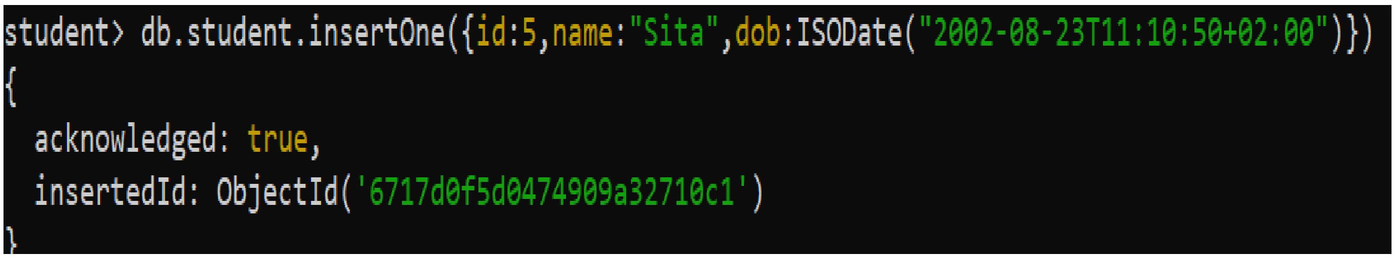


```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Please enter a MongoDB connection string (Default: mongodb://localhost/):

Current Mongosh Log ID: 671902fa7c4a8a27312710bb
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.0
Using MongoDB:          7.0.14
Using Mongosh:          2.3.0
mongosh 2.3.2 is available for download: https://www.mongodb.com/try/download/shell

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

------
   The server generated these startup warnings when booting
   2024-10-23T10:53:07.504+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
------

test> use student
switched to db student
student> 
```

Q2. Insert the following documents in student (id, name, dob)

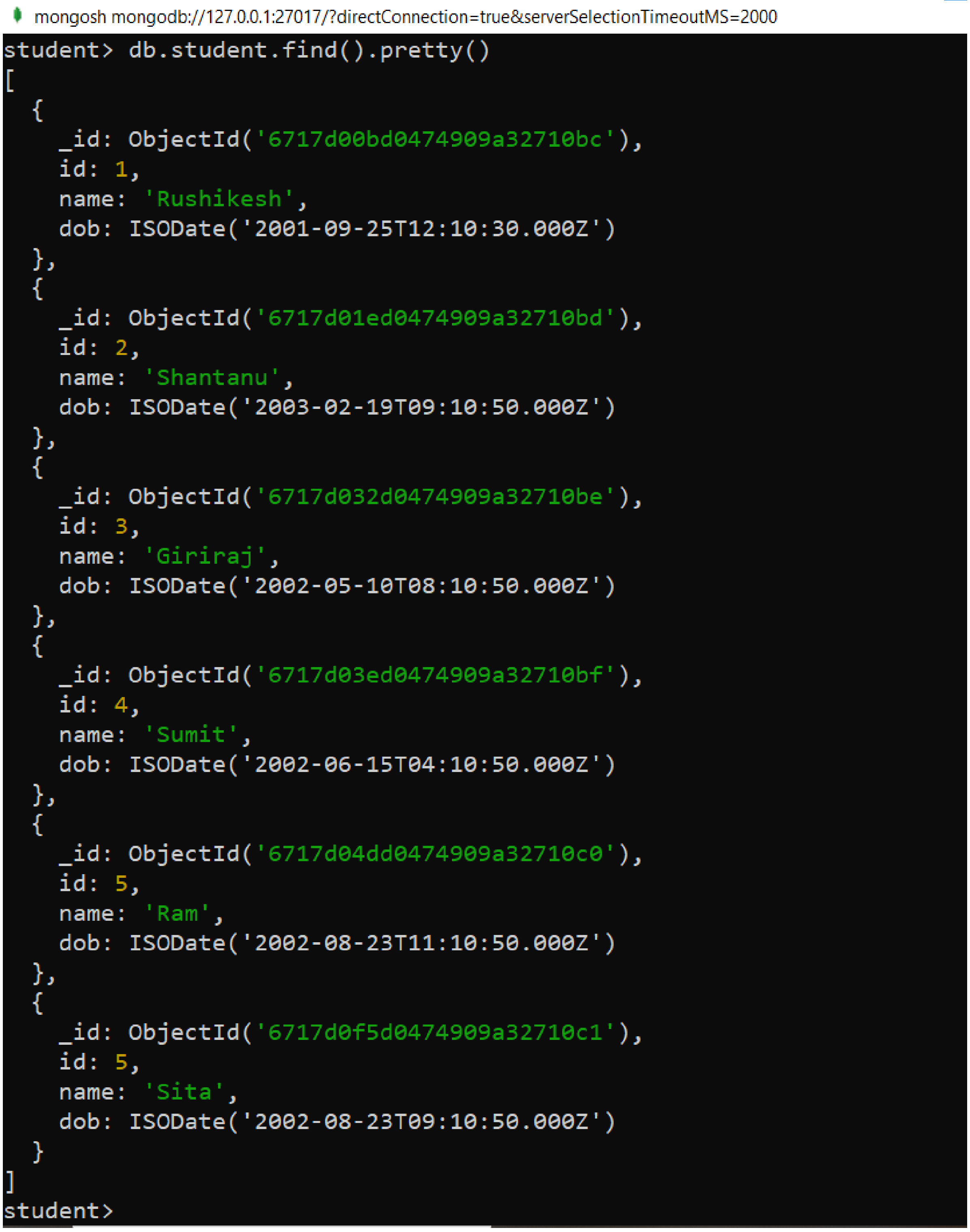


```
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
   2024-10-17T11:04:21.821+05:30: Access control is not enabled for the database. Read and w
------

test> use student
switched to db student
student> db.student.insertOne({id:1,name:"Rushikesh",dob:ISODate("2001-09-25T12:10:30Z")})
{
  acknowledged: true,
  insertedId: ObjectId('6717d00bd0474909a32710bc')
}
student> db.student.insertOne({id:2,name:"Shantanu",dob:ISODate("2003-02-19T09:10:50Z")})
{
  acknowledged: true,
  insertedId: ObjectId('6717d01ed0474909a32710bd')
}
student> db.student.insertOne({id:3,name:"Giriraj",dob:ISODate("2002-05-10T08:10:50Z")})
{
  acknowledged: true,
  insertedId: ObjectId('6717d032d0474909a32710be')
}
student> db.student.insertOne({id:4,name:"Sumit",dob:ISODate("2002-06-15T04:10:50Z")})
{
  acknowledged: true,
  insertedId: ObjectId('6717d03ed0474909a32710bf')
}
student> db.student.insertOne({id:5,name:"Ram",dob:ISODate("2002-08-23T11:10:50Z")})
{
student>
  insertedId: ObjectId('6717d04dd0474909a32710c0')
}
```



```
student> db.student.insertOne({id:5,name:"Sita",dob:ISODate("2002-08-23T11:10:50+02:00")})
{
  acknowledged: true,
  insertedId: ObjectId('6717d0f5d0474909a32710c1')
}
```

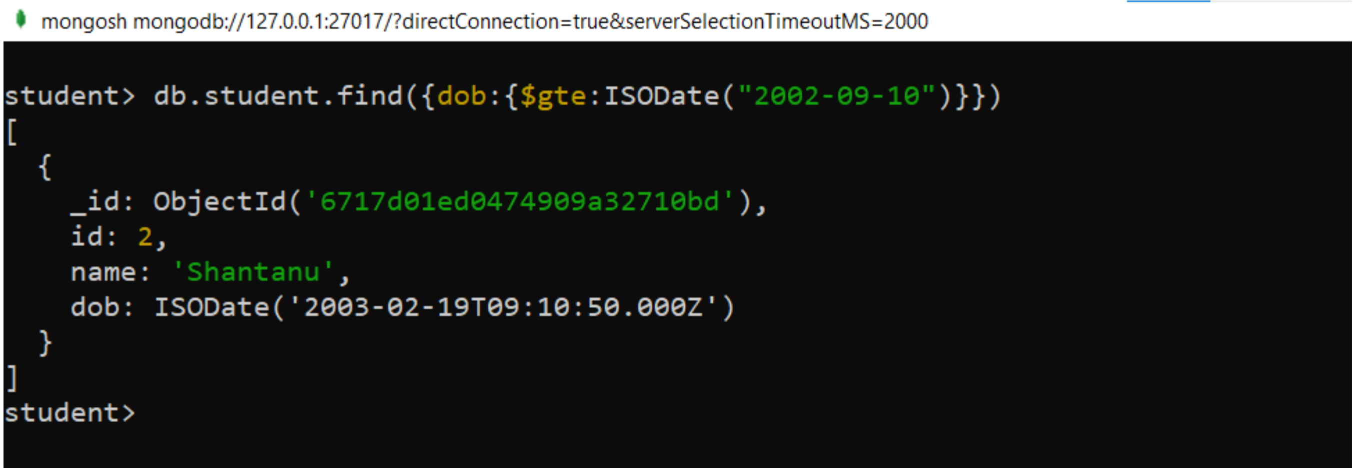
Q3. Print all the documents.



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
student> db.student.find().pretty()
[
  {
    _id: ObjectId('6717d00bd0474909a32710bc'),
    id: 1,
    name: 'Rushikesh',
    dob: ISODate('2001-09-25T12:10:30.000Z')
  },
  {
    _id: ObjectId('6717d01ed0474909a32710bd'),
    id: 2,
    name: 'Shantanu',
    dob: ISODate('2003-02-19T09:10:50.000Z')
  },
  {
    _id: ObjectId('6717d032d0474909a32710be'),
    id: 3,
    name: 'Giriraj',
    dob: ISODate('2002-05-10T08:10:50.000Z')
  },
  {
    _id: ObjectId('6717d03ed0474909a32710bf'),
    id: 4,
    name: 'Sumit',
    dob: ISODate('2002-06-15T04:10:50.000Z')
  },
  {
    _id: ObjectId('6717d04dd0474909a32710c0'),
    id: 5,
    name: 'Ram',
    dob: ISODate('2002-08-23T11:10:50.000Z')
  },
  {
    _id: ObjectId('6717d0f5d0474909a32710c1'),
    id: 5,
    name: 'Sita',
    dob: ISODate('2002-08-23T09:10:50.000Z')
  }
]
student>
```

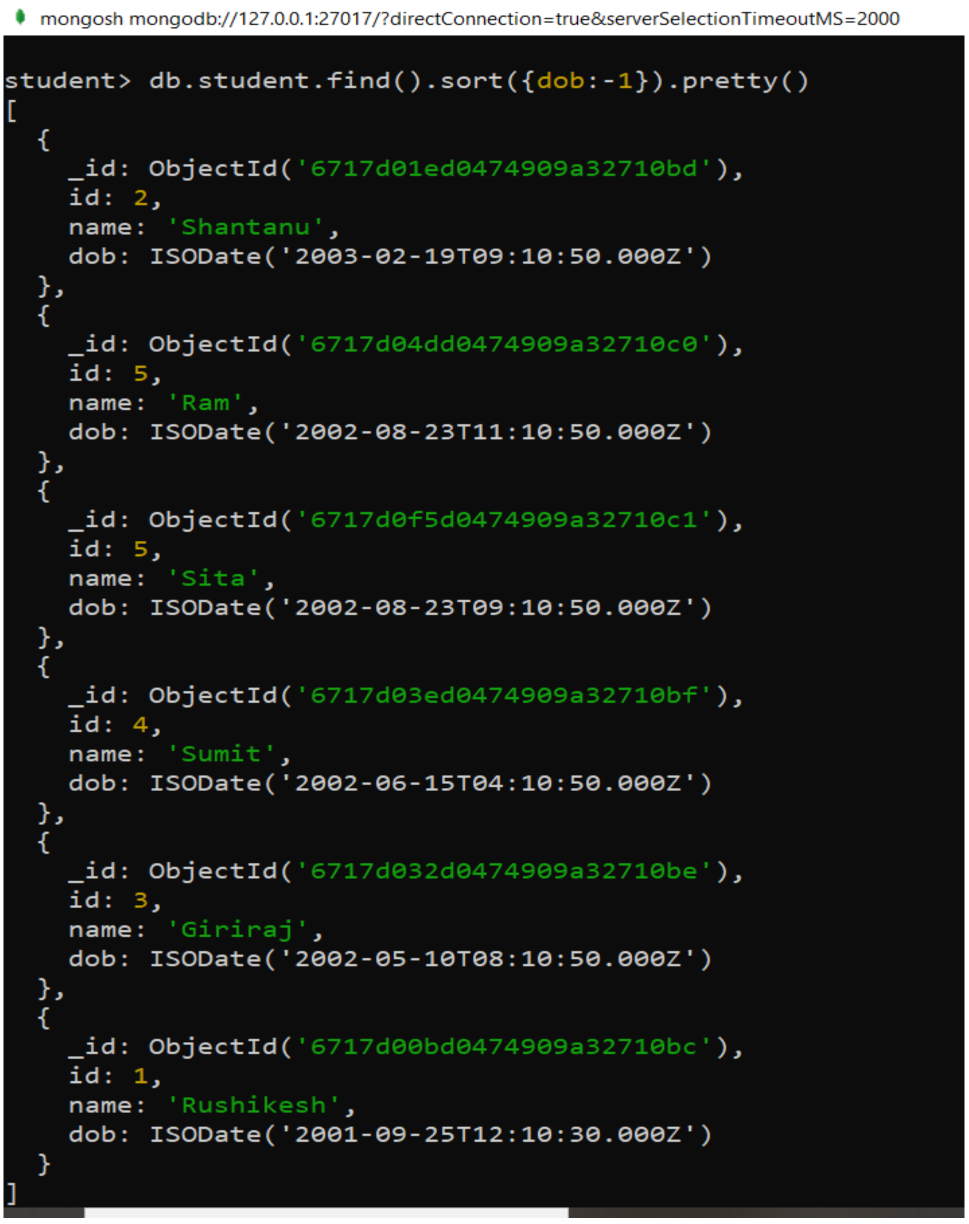
Q4. Find documents in student greater than given data.



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
student> db.student.find({dob:{$gte:ISODate("2002-09-10")}})
[
  {
    _id: ObjectId('6717d01ed0474909a32710bd'),
    id: 2,
    name: 'Shantanu',
    dob: ISODate('2003-02-19T09:10:50.000Z')
  }
]
student>
```

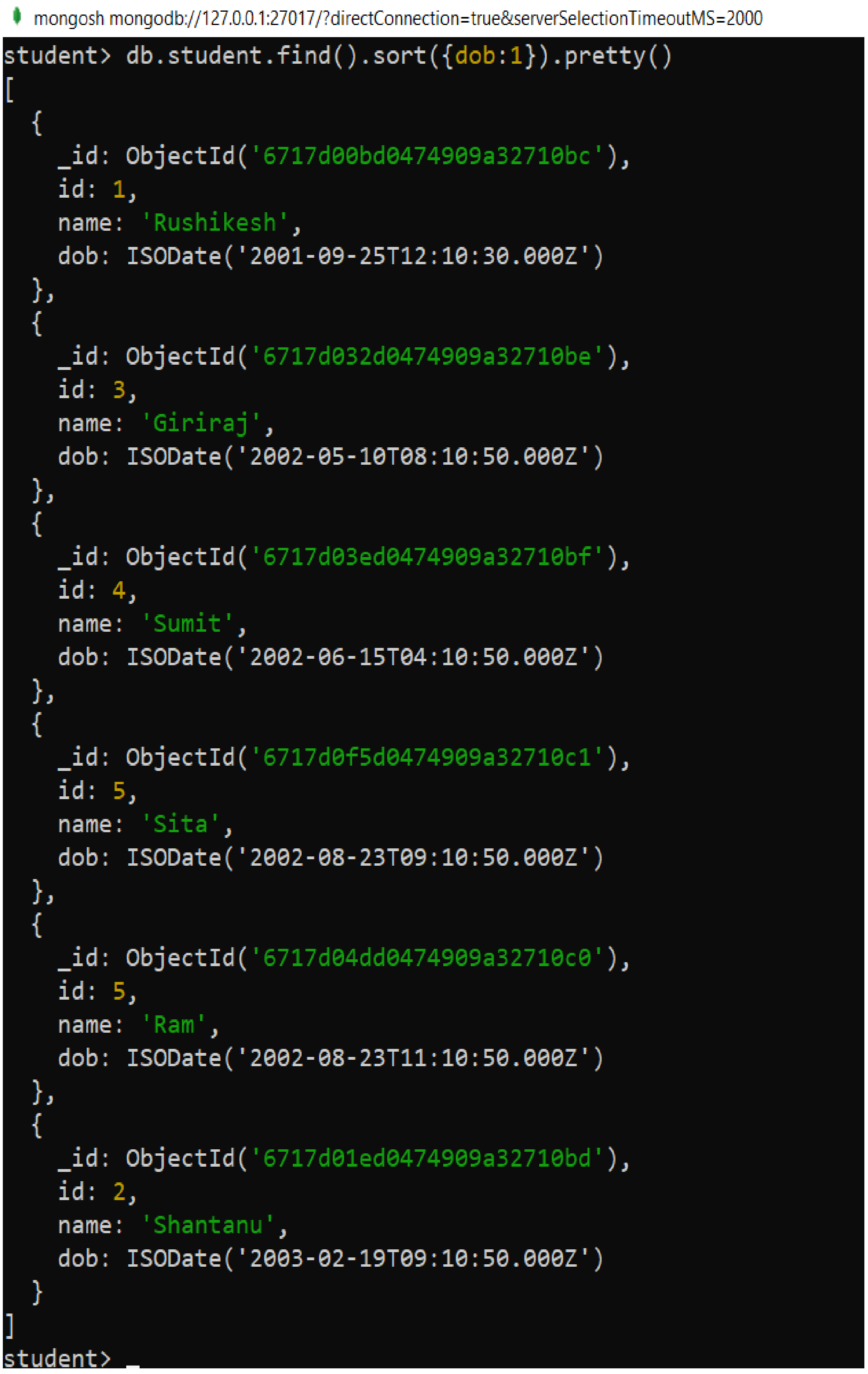
Q5. Sort the documents from student collection in descending order.



mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
student> db.student.find().sort({dob:-1}).pretty()
[
  {
    _id: ObjectId('6717d01ed0474909a32710bd'),
    id: 2,
    name: 'Shantanu',
    dob: ISODate('2003-02-19T09:10:50.000Z')
  },
  {
    _id: ObjectId('6717d04dd0474909a32710c0'),
    id: 5,
    name: 'Ram',
    dob: ISODate('2002-08-23T11:10:50.000Z')
  },
  {
    _id: ObjectId('6717d0f5d0474909a32710c1'),
    id: 5,
    name: 'Sita',
    dob: ISODate('2002-08-23T09:10:50.000Z')
  },
  {
    _id: ObjectId('6717d03ed0474909a32710bf'),
    id: 4,
    name: 'Sumit',
    dob: ISODate('2002-06-15T04:10:50.000Z')
  },
  {
    _id: ObjectId('6717d032d0474909a32710be'),
    id: 3,
    name: 'Giriraj',
    dob: ISODate('2002-05-10T08:10:50.000Z')
  },
  {
    _id: ObjectId('6717d00bd0474909a32710bc'),
    id: 1,
    name: 'Rushikesh',
    dob: ISODate('2001-09-25T12:10:30.000Z')
  }
]
```

Q6. Sort the documents from student collection in ascending order.



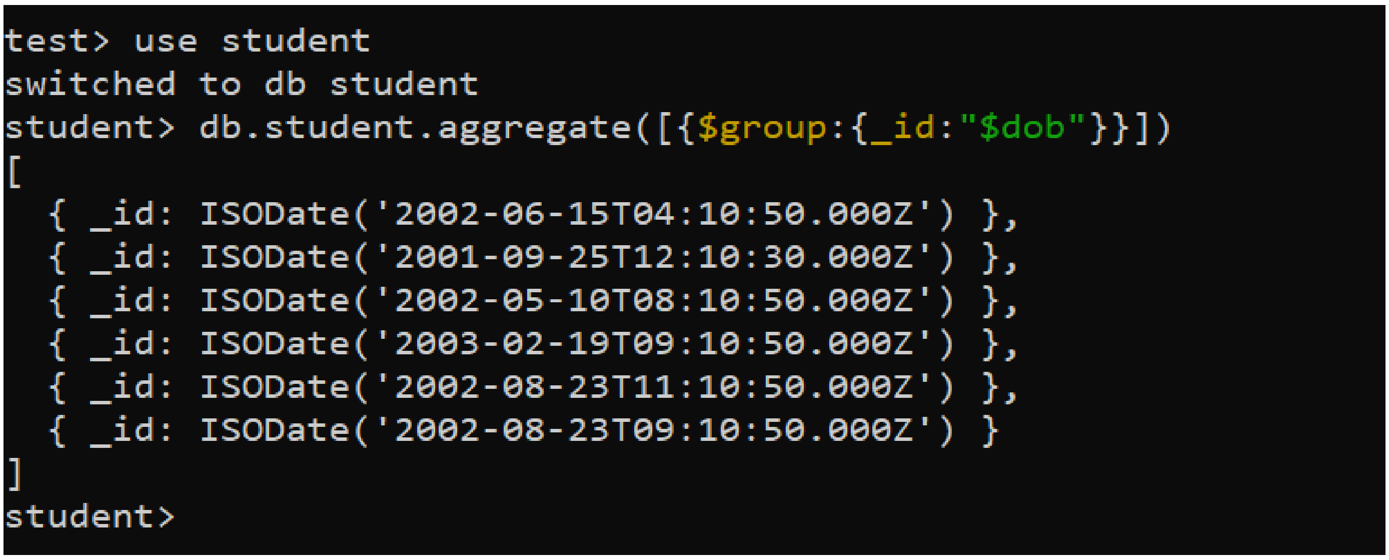
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
student> db.student.find().sort({dob:1}).pretty()
[
  {
    _id: ObjectId('6717d00bd0474909a32710bc'),
    id: 1,
    name: 'Rushikesh',
    dob: ISODate('2001-09-25T12:10:30.000Z')
  },
  {
    _id: ObjectId('6717d032d0474909a32710be'),
    id: 3,
    name: 'Giriraj',
    dob: ISODate('2002-05-10T08:10:50.000Z')
  },
  {
    _id: ObjectId('6717d03ed0474909a32710bf'),
    id: 4,
    name: 'Sumit',
    dob: ISODate('2002-06-15T04:10:50.000Z')
  },
  {
    _id: ObjectId('6717d0f5d0474909a32710c1'),
    id: 5,
    name: 'Sita',
    dob: ISODate('2002-08-23T09:10:50.000Z')
  },
  {
    _id: ObjectId('6717d04dd0474909a32710c0'),
    id: 5,
    name: 'Ram',
    dob: ISODate('2002-08-23T11:10:50.000Z')
  },
  {
    _id: ObjectId('6717d01ed0474909a32710bd'),
    id: 2,
    name: 'Shantanu',
    dob: ISODate('2003-02-19T09:10:50.000Z')
  }
]
student> _
```

EXPERIMENT NO: 8

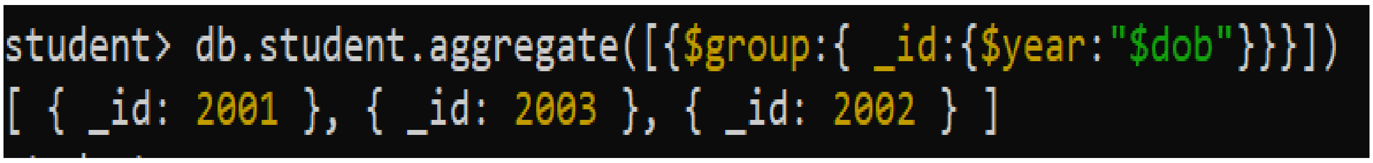
Title:- Implement data queries using aggregate function in MongoDB

Q1. Implement dob document in student.



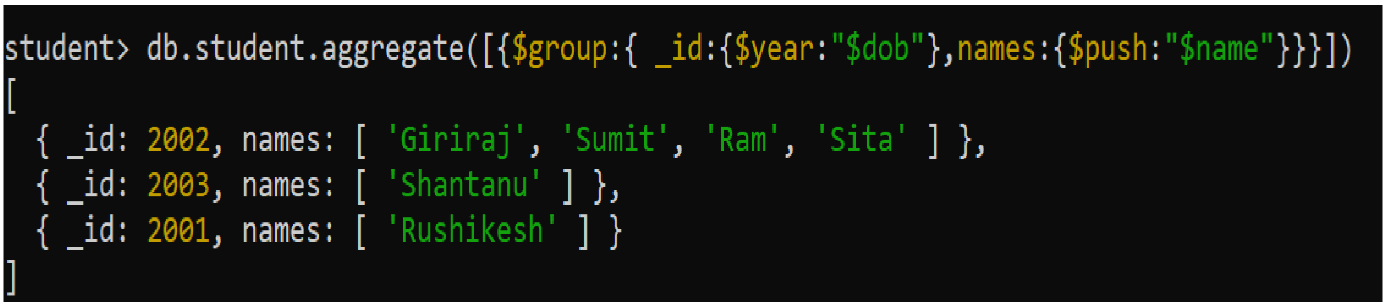
```
test> use student
switched to db student
student> db.student.aggregate([{$group:{_id:"$dob"}}])
[
  { _id: ISODate('2002-06-15T04:10:50.000Z') },
  { _id: ISODate('2001-09-25T12:10:30.000Z') },
  { _id: ISODate('2002-05-10T08:10:50.000Z') },
  { _id: ISODate('2003-02-19T09:10:50.000Z') },
  { _id: ISODate('2002-08-23T11:10:50.000Z') },
  { _id: ISODate('2002-08-23T09:10:50.000Z') }
]
student>
```

Q2. Implement year document from student.



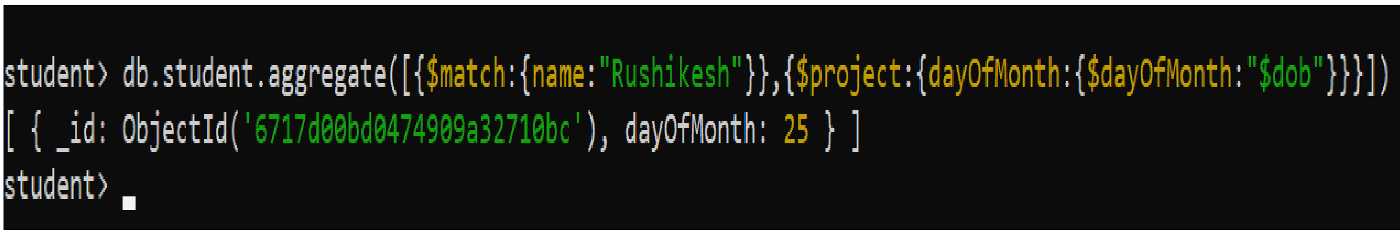
```
student> db.student.aggregate([{$group:{ _id:{$year:"$dob"}}}])
[ { _id: 2001 }, { _id: 2003 }, { _id: 2002 } ]
```

Q3. Find name documents.



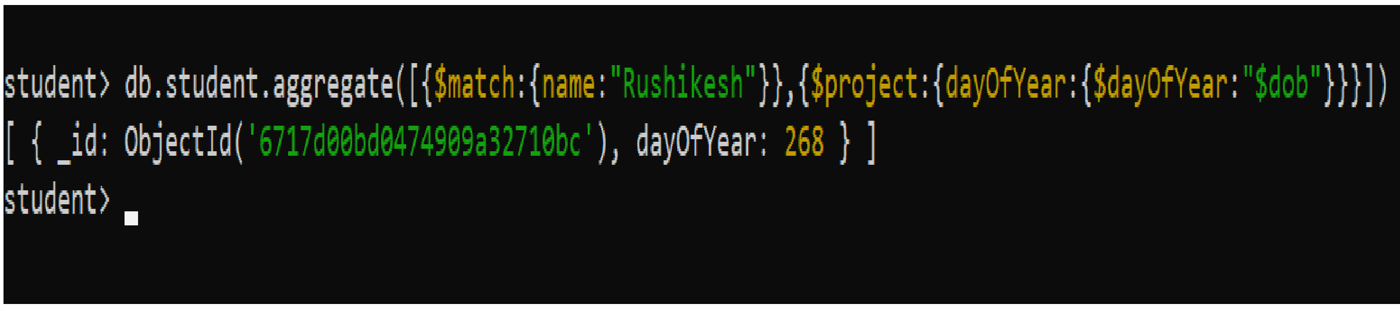
```
student> db.student.aggregate([{$group:{ _id:{$year:"$dob"},names:{$push:"$name"}}}])
[
  { _id: 2002, names: [ 'Giriraj', 'Sumit', 'Ram', 'Sita' ] },
  { _id: 2003, names: [ 'Shantanu' ] },
  { _id: 2001, names: [ 'Rushikesh' ] }
]
```

Q4. Implement day of month from student.



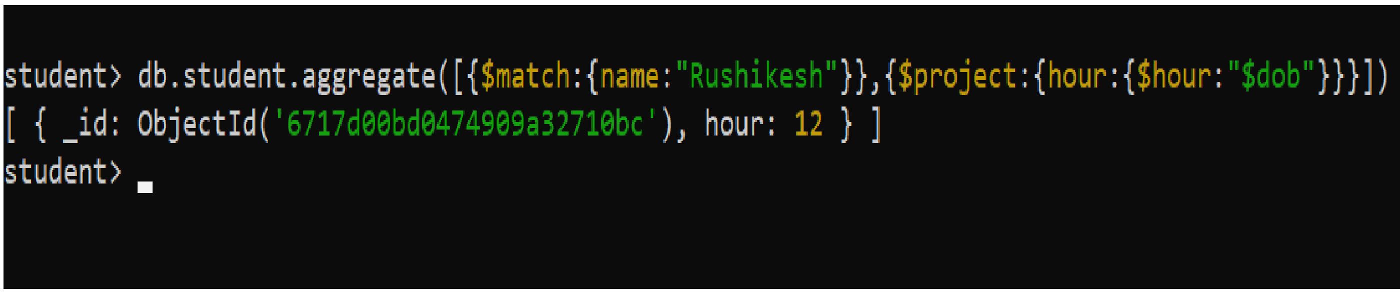
```
student> db.student.aggregate([{$match:{name:"Rushikesh"}},{$project:{dayOfMonth:{$dayOfMonth:"$dob"}}}])
[ { _id: ObjectId('6717d00bd0474909a32710bc'), dayOfMonth: 25 } ]
student>
```

Q5. Implement day of year from student.



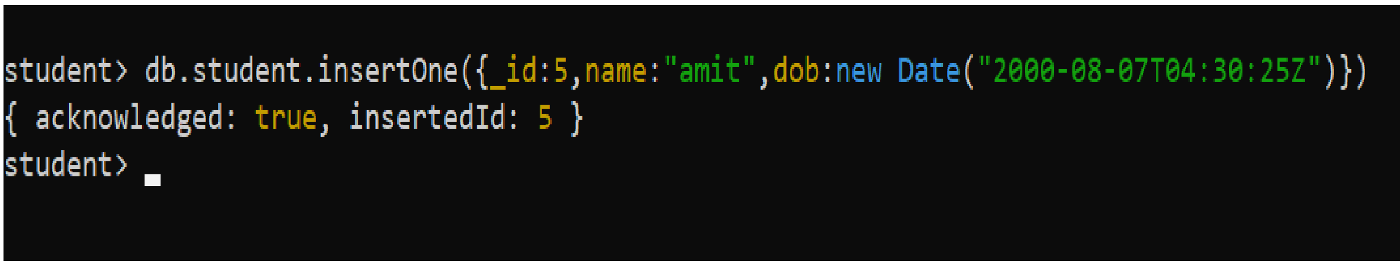
```
student> db.student.aggregate([{$match:{name:"Rushikesh"}},{$project:{dayOfYear:{$dayOfYear:"$dob"}}}])
[ { _id: ObjectId('6717d00bd0474909a32710bc'), dayOfYear: 268 } ]
student> 
```

Q6. Implement hours of day from student.



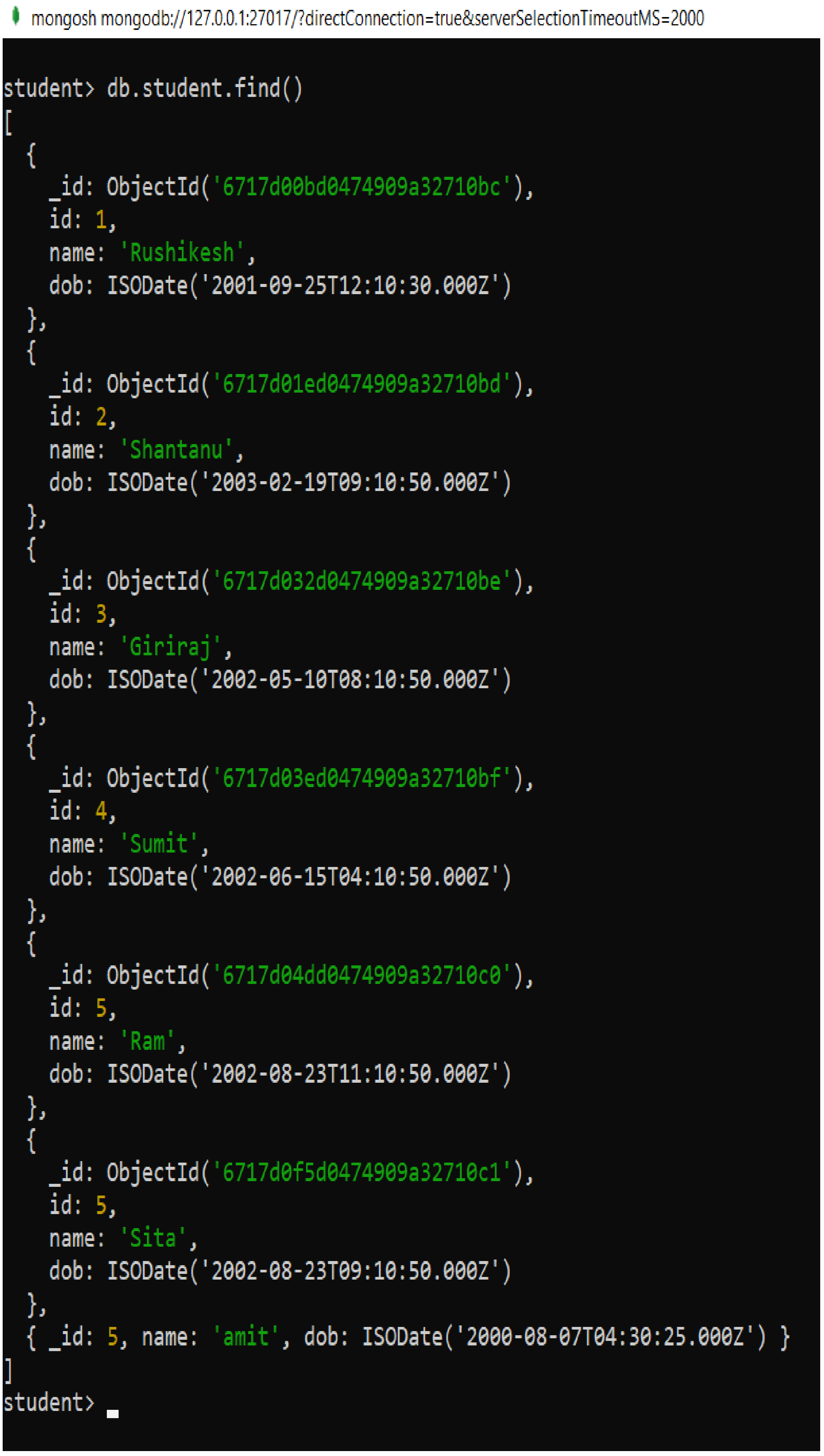
```
student> db.student.aggregate([{$match:{name:"Rushikesh"}},{$project:{hour:{$hour:"$dob"}}}])
[ { _id: ObjectId('6717d00bd0474909a32710bc'), hour: 12 } ]
student> 
```

Q7. Insert new document using new Date function with id 5 and name amit.



```
student> db.student.insertOne({_id:5,name:"amit",dob:new Date("2000-08-07T04:30:25Z")})
{ acknowledged: true, insertedId: 5 }
student> 
```

Q8. Print all documents.

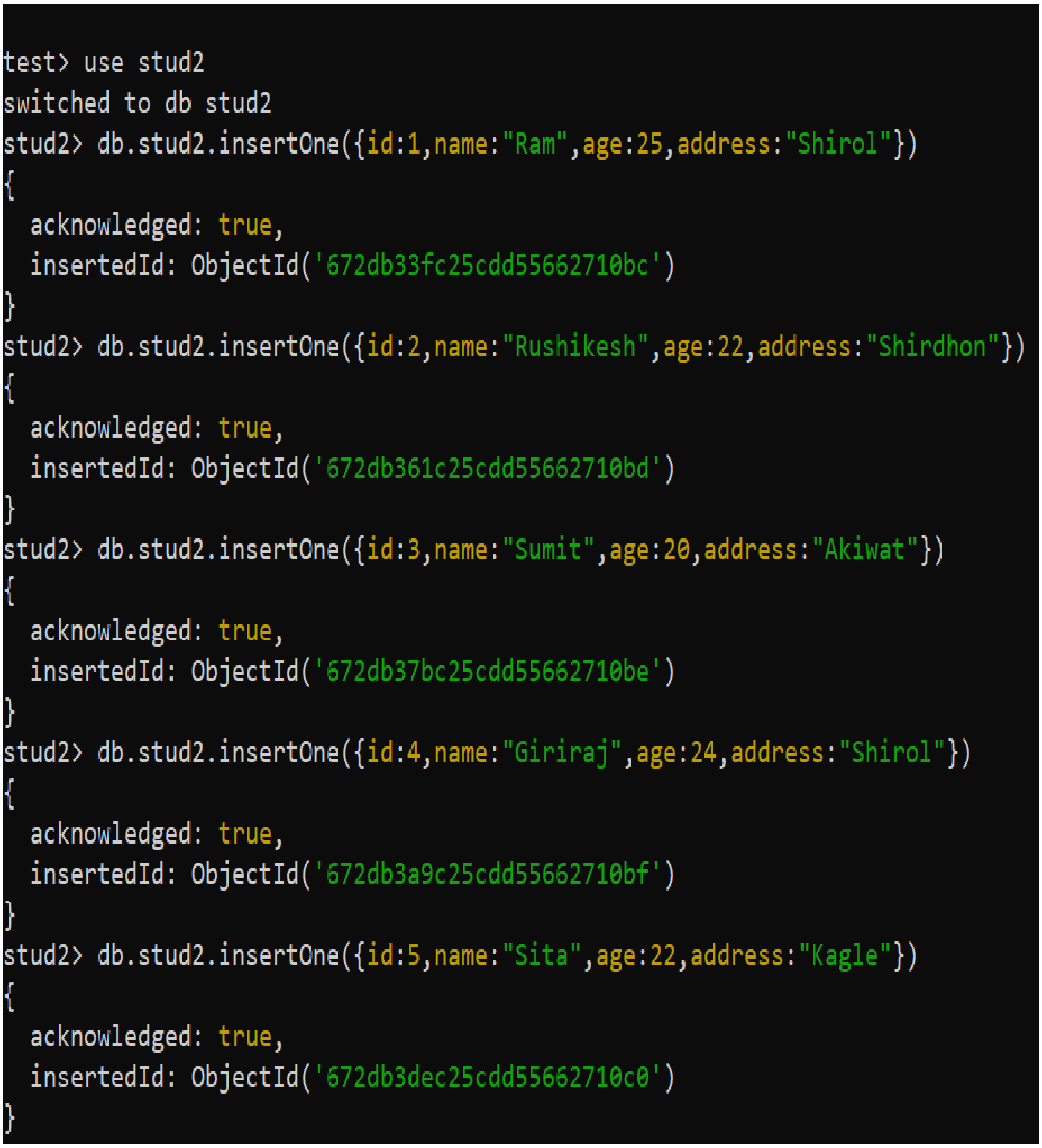

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000

```
student> db.student.find()
[
  {
    _id: ObjectId('6717d00bd0474909a32710bc'),
    id: 1,
    name: 'Rushikesh',
    dob: ISODate('2001-09-25T12:10:30.000Z')
  },
  {
    _id: ObjectId('6717d01ed0474909a32710bd'),
    id: 2,
    name: 'Shantanu',
    dob: ISODate('2003-02-19T09:10:50.000Z')
  },
  {
    _id: ObjectId('6717d032d0474909a32710be'),
    id: 3,
    name: 'Giriraj',
    dob: ISODate('2002-05-10T08:10:50.000Z')
  },
  {
    _id: ObjectId('6717d03ed0474909a32710bf'),
    id: 4,
    name: 'Sumit',
    dob: ISODate('2002-06-15T04:10:50.000Z')
  },
  {
    _id: ObjectId('6717d04dd0474909a32710c0'),
    id: 5,
    name: 'Ram',
    dob: ISODate('2002-08-23T11:10:50.000Z')
  },
  {
    _id: ObjectId('6717d0f5d0474909a32710c1'),
    id: 5,
    name: 'Sita',
    dob: ISODate('2002-08-23T09:10:50.000Z')
  },
  { _id: 5, name: 'amit', dob: ISODate('2000-08-07T04:30:25.000Z') }
]
student> _
```

EXPERIMENT NO: 9

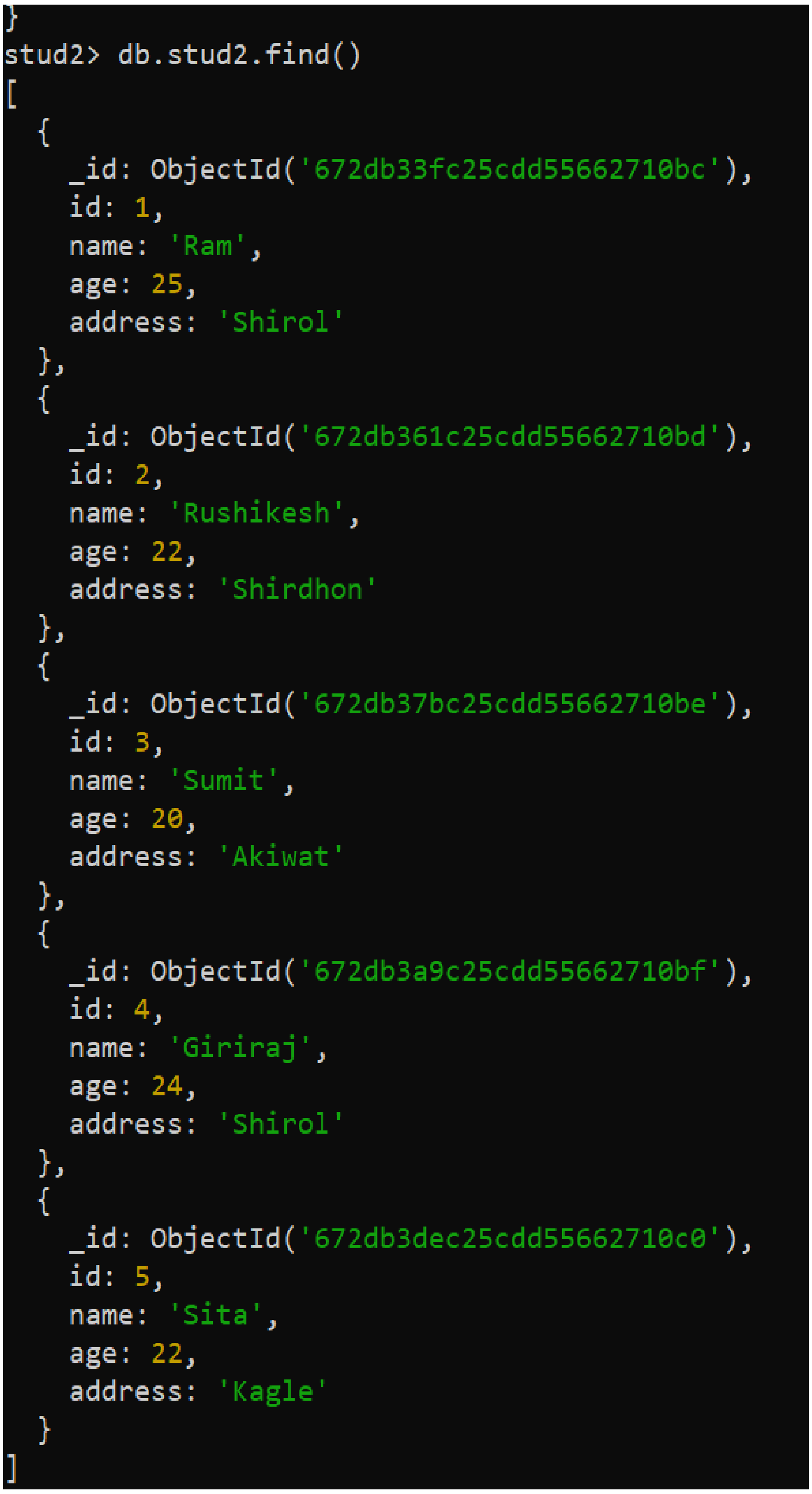
Title: - Embedded documents in MongoDB (Nested Document).

Q1. Create database students (id, name, age, address).



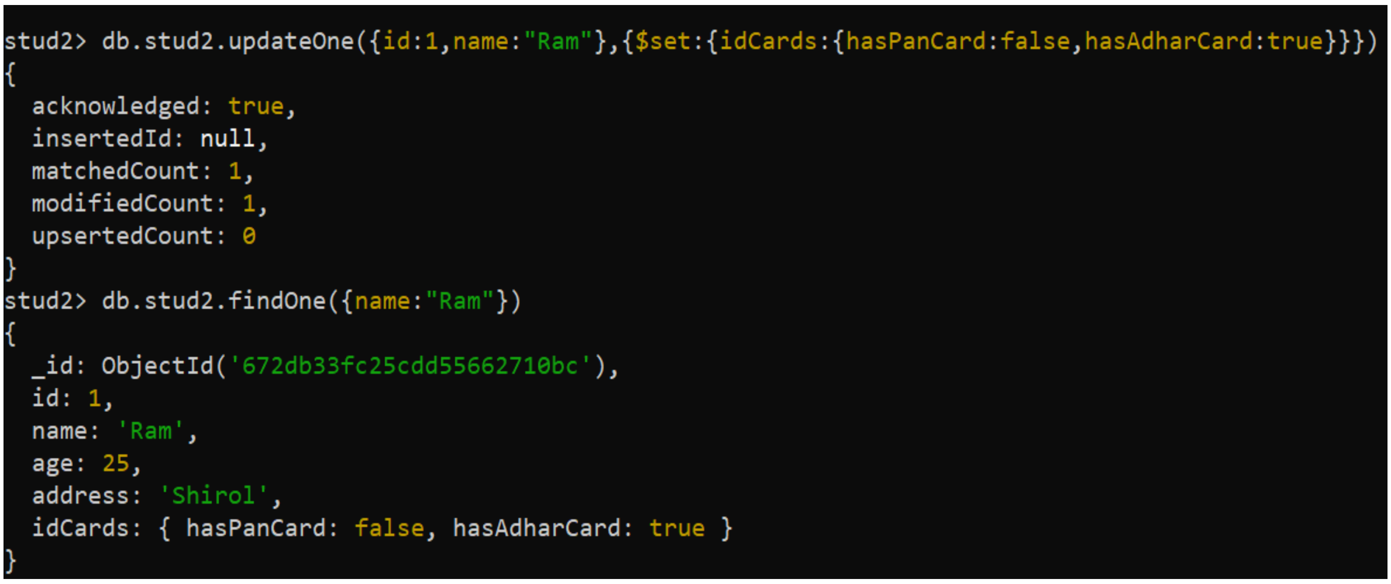
```
test> use stud2
switched to db stud2
stud2> db.stud2.insertOne({id:1,name:"Ram",age:25,address:"Shirol"})
{
  acknowledged: true,
  insertedId: ObjectId('672db33fc25cdd55662710bc')
}
stud2> db.stud2.insertOne({id:2,name:"Rushikesh",age:22,address:"Shirdhon"})
{
  acknowledged: true,
  insertedId: ObjectId('672db361c25cdd55662710bd')
}
stud2> db.stud2.insertOne({id:3,name:"Sumit",age:20,address:"Akiwat"})
{
  acknowledged: true,
  insertedId: ObjectId('672db37bc25cdd55662710be')
}
stud2> db.stud2.insertOne({id:4,name:"Giriraj",age:24,address:"Shirol"})
{
  acknowledged: true,
  insertedId: ObjectId('672db3a9c25cdd55662710bf')
}
stud2> db.stud2.insertOne({id:5,name:"Sita",age:22,address:"Kagle"})
{
  acknowledged: true,
  insertedId: ObjectId('672db3dec25cdd55662710c0')
}
```

Print all documents from collection students.



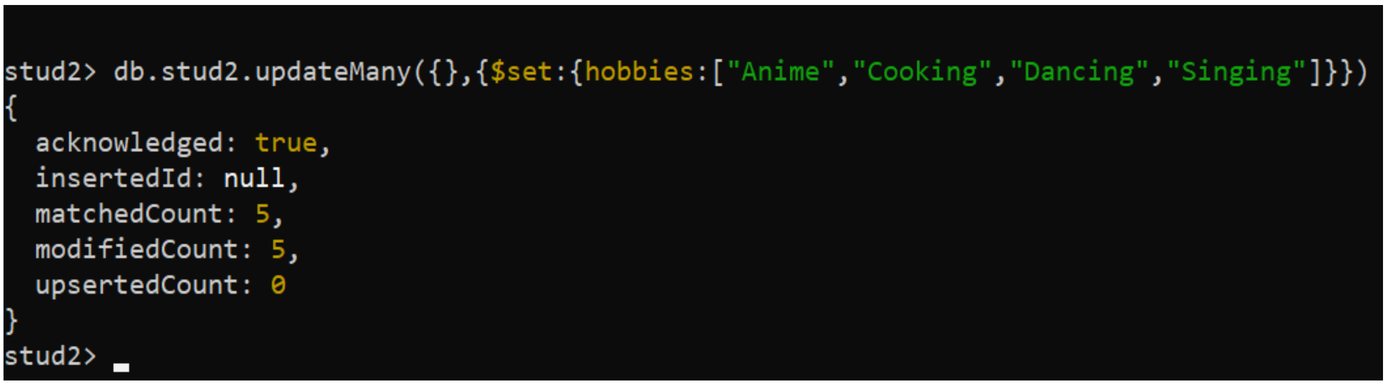
```
}
stud2> db.stud2.find()
[
  {
    _id: ObjectId('672db33fc25cdd55662710bc'),
    id: 1,
    name: 'Ram',
    age: 25,
    address: 'Shirol'
  },
  {
    _id: ObjectId('672db361c25cdd55662710bd'),
    id: 2,
    name: 'Rushikesh',
    age: 22,
    address: 'Shirdhon'
  },
  {
    _id: ObjectId('672db37bc25cdd55662710be'),
    id: 3,
    name: 'Sumit',
    age: 20,
    address: 'Akiwat'
  },
  {
    _id: ObjectId('672db3a9c25cdd55662710bf'),
    id: 4,
    name: 'Giriraj',
    age: 24,
    address: 'Shirol'
  },
  {
    _id: ObjectId('672db3dec25cdd55662710c0'),
    id: 5,
    name: 'Sita',
    age: 22,
    address: 'Kagle'
  }
]
```

Q 2. Update the document of collection students with id Cards (Pan Card has false & Adhar Card has true) where id: 1 & name: Ram.



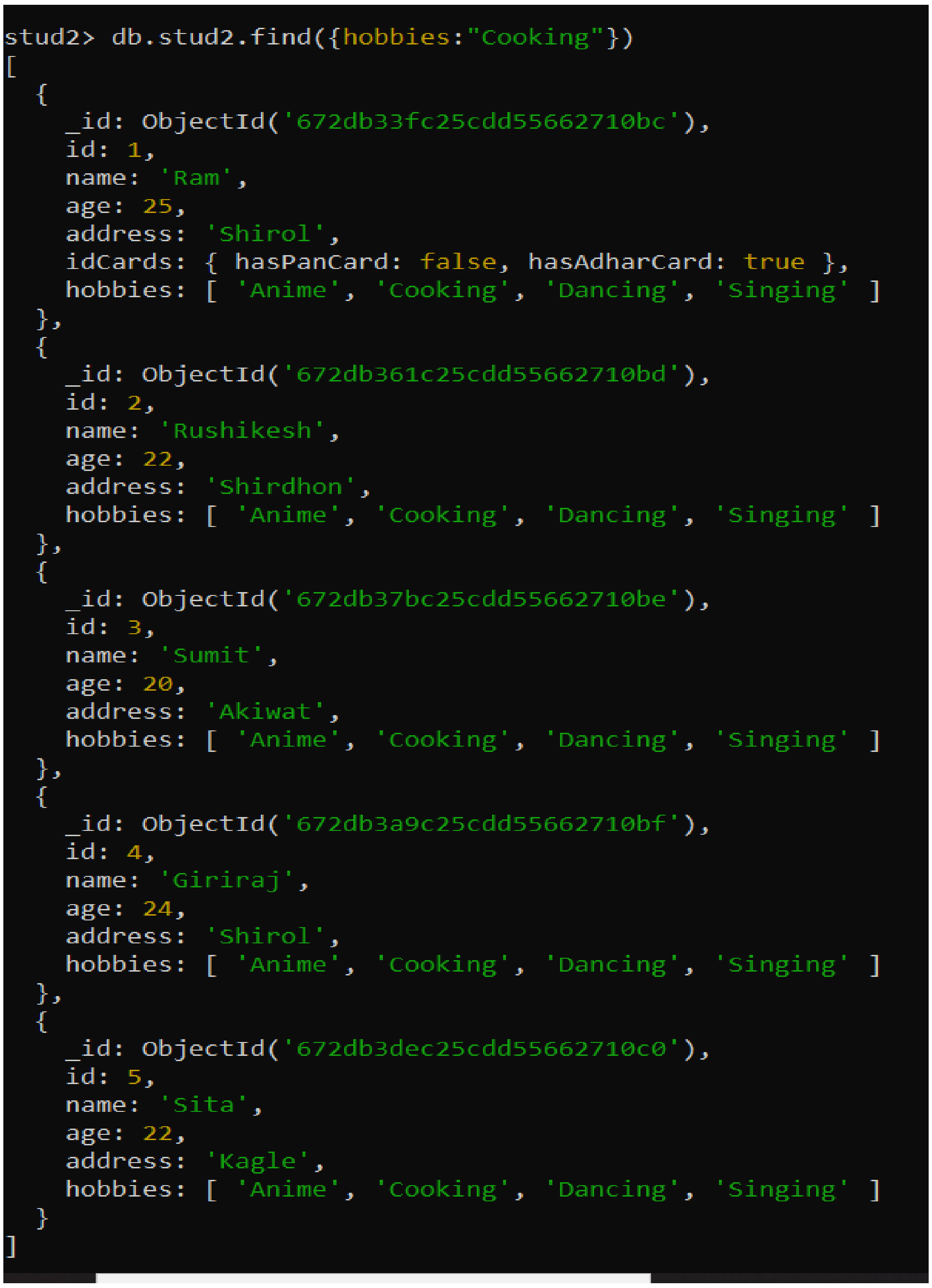
```
stud2> db.stud2.updateOne({id:1,name:"Ram"},{$set:{idCards:{hasPanCard:false,hasAdharCard:true}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
stud2> db.stud2.findOne({name:"Ram"})
{
  _id: ObjectId('672db33fc25cdd55662710bc'),
  id: 1,
  name: 'Ram',
  age: 25,
  address: 'Shirol',
  idCards: { hasPanCard: false, hasAdharCard: true }
}
```

Q3. Update the document of collection students with hobbies (Anime, cooking dancing, and singing).



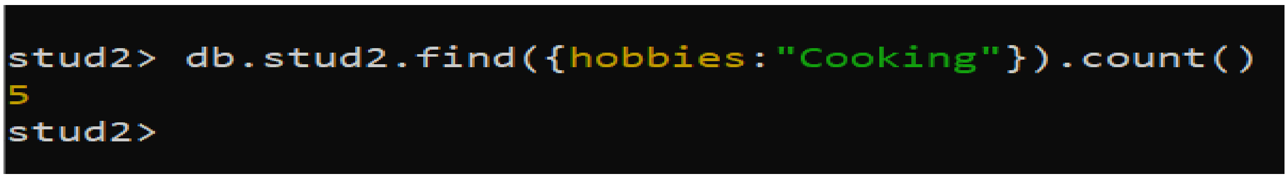
```
stud2> db.stud2.updateMany({},{$set:{hobbies:["Anime","Cooking","Dancing","Singing"]}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}
stud2> _
```

Q4. Find the document from students where hobbies are cooking



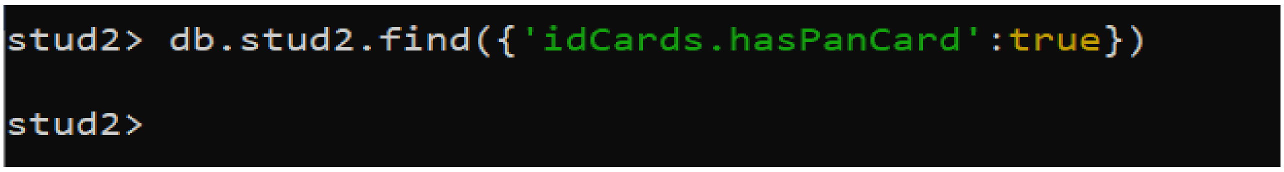
```
stud2> db.stud2.find({hobbies:"Cooking"})
[
  {
    _id: ObjectId('672db33fc25cdd55662710bc'),
    id: 1,
    name: 'Ram',
    age: 25,
    address: 'Shirol',
    idCards: { hasPanCard: false, hasAdharCard: true },
    hobbies: [ 'Anime', 'Cooking', 'Dancing', 'Singing' ]
  },
  {
    _id: ObjectId('672db361c25cdd55662710bd'),
    id: 2,
    name: 'Rushikesh',
    age: 22,
    address: 'Shirdhon',
    hobbies: [ 'Anime', 'Cooking', 'Dancing', 'Singing' ]
  },
  {
    _id: ObjectId('672db37bc25cdd55662710be'),
    id: 3,
    name: 'Sumit',
    age: 20,
    address: 'Akiwat',
    hobbies: [ 'Anime', 'Cooking', 'Dancing', 'Singing' ]
  },
  {
    _id: ObjectId('672db3a9c25cdd55662710bf'),
    id: 4,
    name: 'Giriraj',
    age: 24,
    address: 'Shirol',
    hobbies: [ 'Anime', 'Cooking', 'Dancing', 'Singing' ]
  },
  {
    _id: ObjectId('672db3dec25cdd55662710c0'),
    id: 5,
    name: 'Sita',
    age: 22,
    address: 'Kagle',
    hobbies: [ 'Anime', 'Cooking', 'Dancing', 'Singing' ]
  }
]
```

Q5. Count the documents from students where hobbies are cooking.



```
stud2> db.stud2.find({hobbies:"Cooking"}).count()
5
stud2>
```

Q6. Find the document from students where id cards (Pan Card has True).

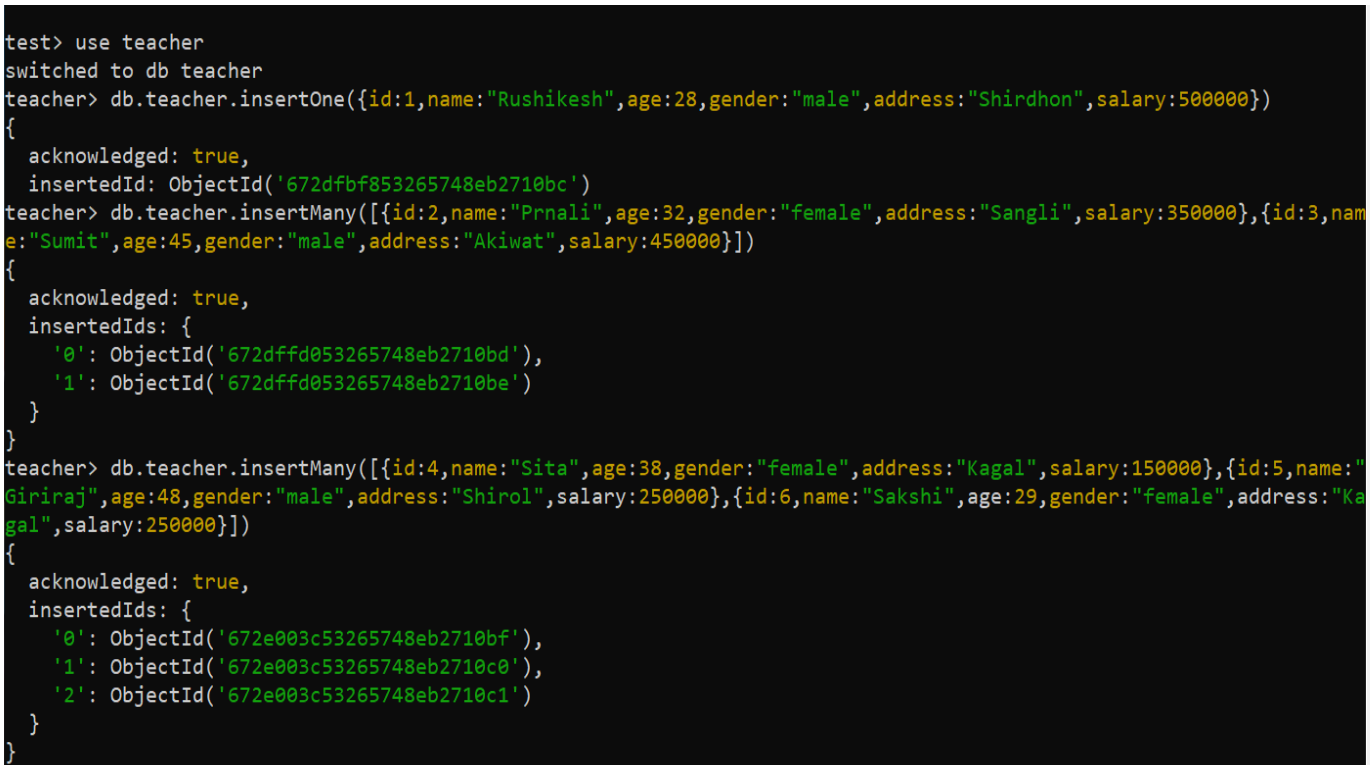


```
stud2> db.stud2.find({'idCards.hasPanCard':true})

stud2>
```

EXPERIMENT NO: 10

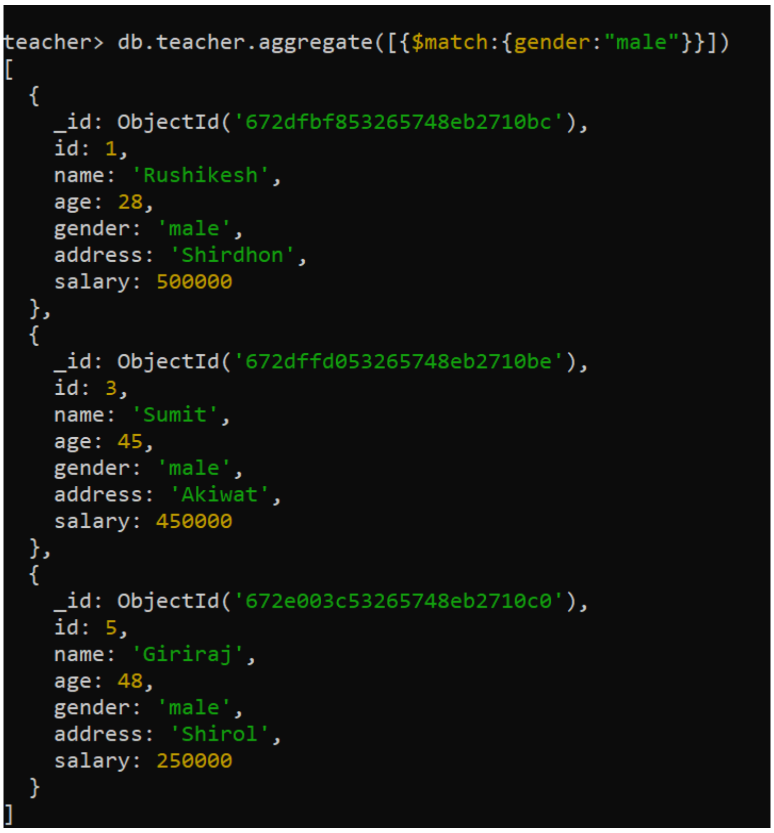
Title: -Bucket operator in MongoDB.

Q1. Create database teacher and insert following documents in collection teacher.



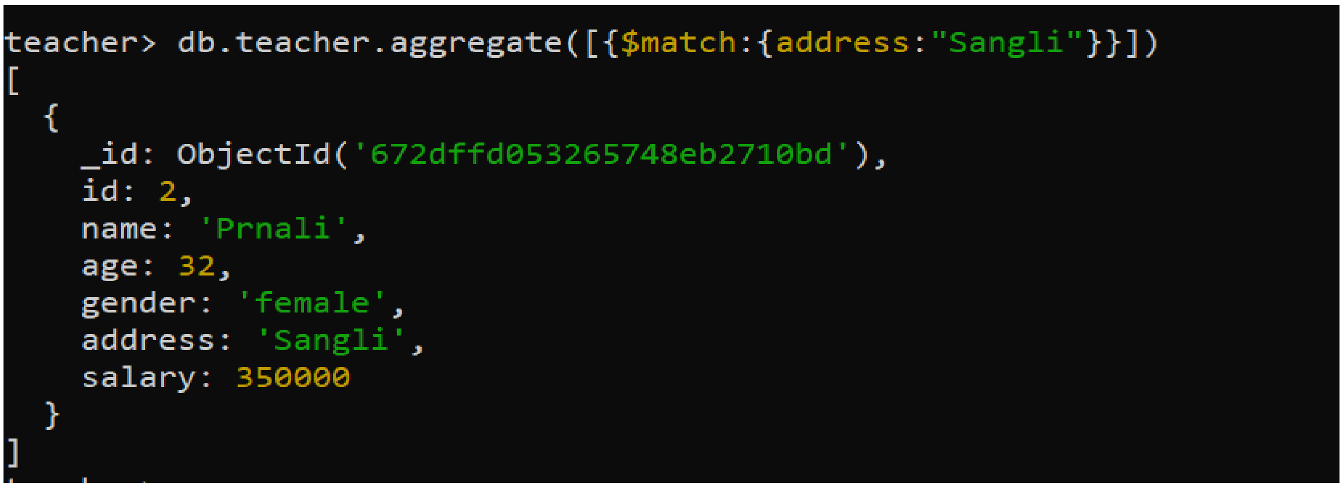
```
test> use teacher
switched to db teacher
teacher> db.teacher.insertOne({id:1,name:"Rushikesh",age:28,gender:"male",address:"Shirdhon",salary:500000})
{
  acknowledged: true,
  insertedId: ObjectId('672dfbf853265748eb2710bc')
teacher> db.teacher.insertMany([{id:2,name:"Prnali",age:32,gender:"female",address:"Sangli",salary:350000},{id:3,nam
e:"Sumit",age:45,gender:"male",address:"Akiwat",salary:450000}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672dffd053265748eb2710bd'),
    '1': ObjectId('672dffd053265748eb2710be')
  }
}
teacher> db.teacher.insertMany([{id:4,name:"Sita",age:38,gender:"female",address:"Kagal",salary:150000},{id:5,name:"
Giriraj",age:48,gender:"male",address:"Shirol",salary:250000},{id:6,name:"Sakshi",age:29,gender:"female",address:"Ka
gal",salary:250000}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('672e003c53265748eb2710bf'),
    '1': ObjectId('672e003c53265748eb2710c0'),
    '2': ObjectId('672e003c53265748eb2710c1')
  }
}
```

Q2. Implement gender = “male” document in teacher.



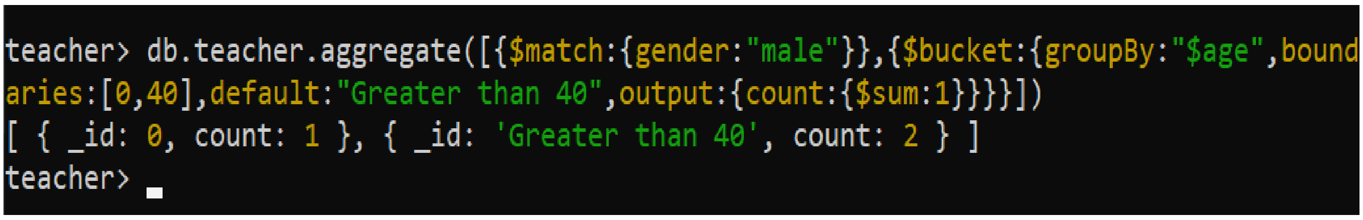
```
teacher> db.teacher.aggregate([{$match:{gender:"male"}}])
[
  {
    _id: ObjectId('672dfbf853265748eb2710bc'),
    id: 1,
    name: 'Rushikesh',
    age: 28,
    gender: 'male',
    address: 'Shirdhon',
    salary: 500000
  },
  {
    _id: ObjectId('672dffd053265748eb2710be'),
    id: 3,
    name: 'Sumit',
    age: 45,
    gender: 'male',
    address: 'Akiwat',
    salary: 450000
  },
  {
    _id: ObjectId('672e003c53265748eb2710c0'),
    id: 5,
    name: 'Giriraj',
    age: 48,
    gender: 'male',
    address: 'Shirol',
    salary: 250000
  }
]
```

Q3. Implement document with address = “Sangli” from teacher.



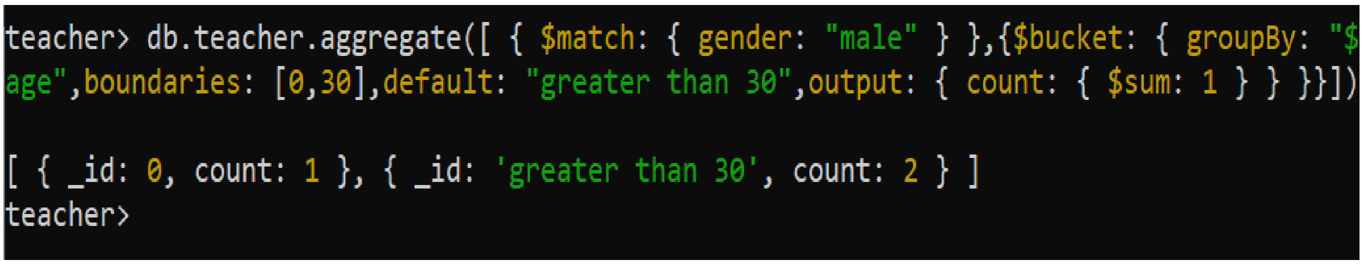
```
teacher> db.teacher.aggregate([{$match:{address:"Sangli"}}])
[
  {
    _id: ObjectId('672dffd053265748eb2710bd'),
    id: 2,
    name: 'Prnali',
    age: 32,
    gender: 'female',
    address: 'Sangli',
    salary: 350000
  }
]
```

Q4. Find male age record greater than 40



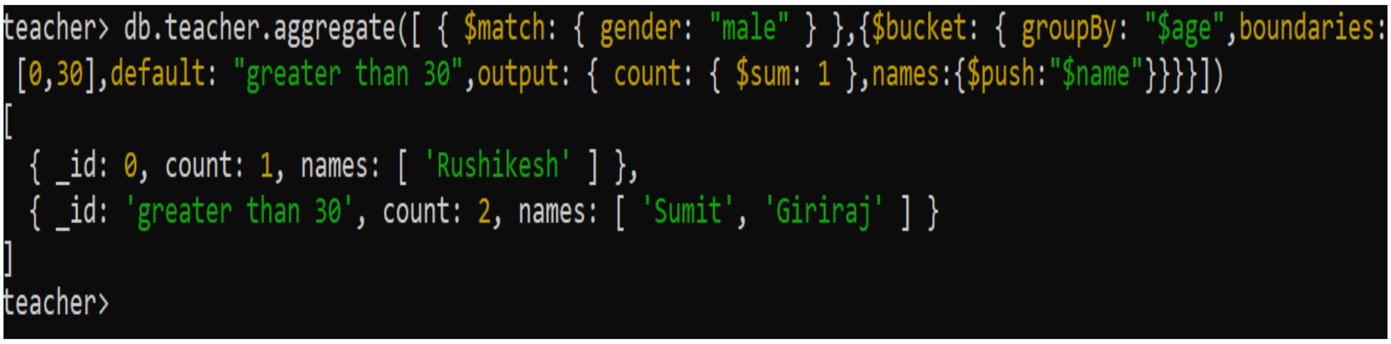
```
teacher> db.teacher.aggregate([{$match:{gender:"male"}},{$bucket:{groupBy:"$age",bound
aries:[0,40],default:"Greater than 40",output:{count:{$sum:1}}}}])
[ { _id: 0, count: 1 }, { _id: 'Greater than 40', count: 2 } ]
teacher>
```

Q5. Find male age record greater than 30



```
teacher> db.teacher.aggregate([ { $match: { gender: "male" } },{$bucket: { groupBy: "$
age",boundaries: [0,30],default: "greater than 30",output: { count: { $sum: 1 } } }}])

[ { _id: 0, count: 1 }, { _id: 'greater than 30', count: 2 } ]
teacher>
```

Q6. Find male age record greater than 30 with their names.



```
teacher> db.teacher.aggregate([ { $match: { gender: "male" } },{$bucket: { groupBy: "$age",boundaries:
 [0,30],default: "greater than 30",output: { count: { $sum: 1 },names:{$push:"$name"}}}}])
[
  { _id: 0, count: 1, names: [ 'Rushikesh' ] },
  { _id: 'greater than 30', count: 2, names: [ 'Sumit', 'Giriraj' ] }
]
teacher>
```