Social Networking System

DBMS Assignment 3

Kartika Nair (PES1UG19CS213)

Krithika Ragothaman (PES1UG19CS231)

Maitreyi P (PES1UG19CS254)

Database Management Systems (UE19CS301)

11 November 2021

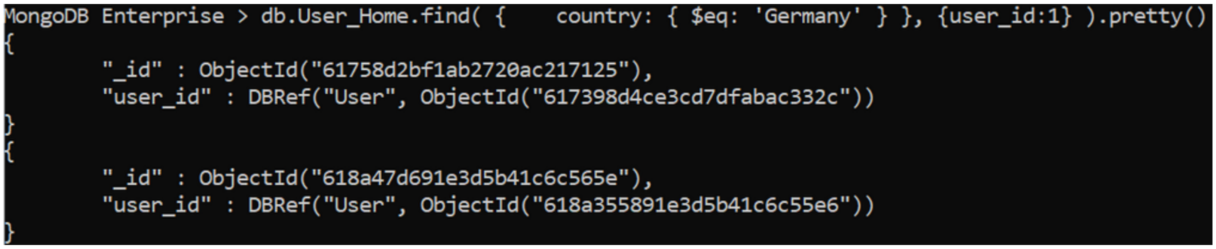
Contents

Chapter 1

Complete Working Model of Database Application

1.1 Simple Queries

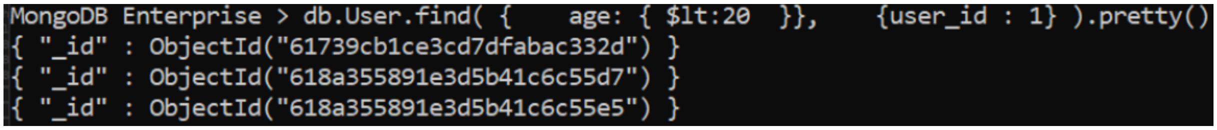
Retrieving details of all users living in Germany,



```
MongoDB Enterprise > db.User_Home.find( {    country: { $eq: 'Germany' } }, {user_id:1} ).pretty()
{
        "_id" : ObjectId("61758d2bf1ab2720ac217125"),
        "user_id" : DBRef("User", ObjectId("617398d4ce3cd7dfabac332c"))
}
{

        "_id" : ObjectId("618a47d691e3d5b41c6c565e"),
        "user_id" : DBRef("User", ObjectId("618a355891e3d5b41c6c55e6"))
}
```

Figure 1.1: Users living in Germany.

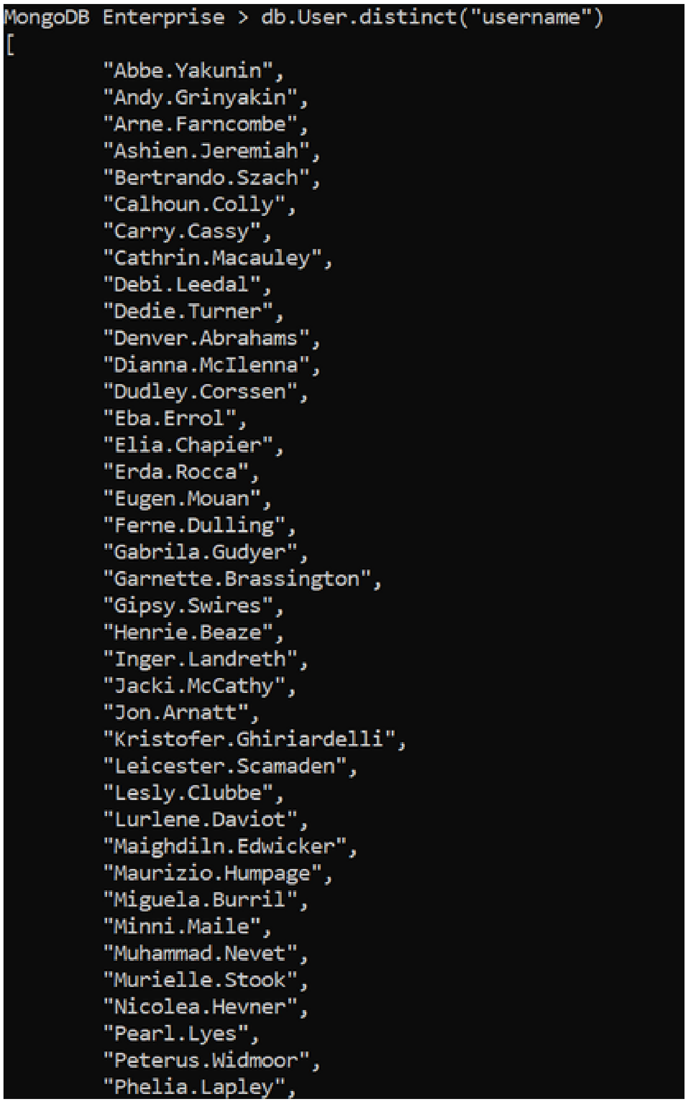
Retrieving details of all users who are under 20 years of age,



```
MongoDB Enterprise > db.User.find( {    age: { $lt:20  }},    {user_id : 1} ).pretty()
{ "_id" : ObjectId("61739cb1ce3cd7dfabac332d") }
{ "_id" : ObjectId("618a355891e3d5b41c6c55d7") }
{ "_id" : ObjectId("618a355891e3d5b41c6c55e5") }
```

Figure 1.2: Users younger than twenty.

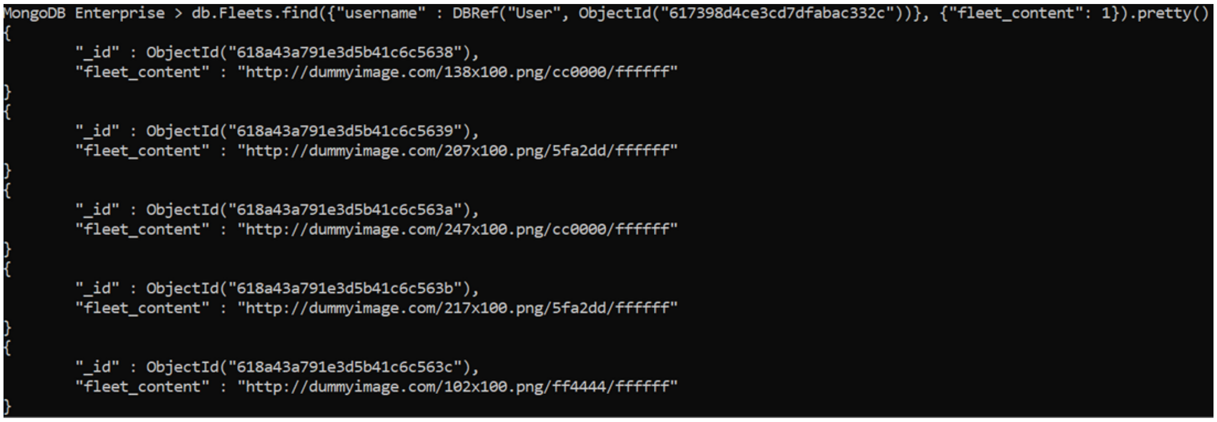
Retrieving all usernames in the Social Network,



```
MongoDB Enterprise > db.User.distinct("username")
[
        "Abbe.Yakunin",
        "Andy.Grinyakin",
        "Arne.Farncombe",
        "Ashien.Jeremiah",
        "Bertrando.Szach",
        "Calhoun.Colly",
        "Carry.Cassy",
        "Cathrin.Macauley",
        "Debi.Leedal",
        "Dedie.Turner",
        "Denver.Abrahams",
        "Dianna.McIlenna",
        "Dudley.Corssen",
        "Eba.Errol",
        "Elia.Chapier",
        "Erda.Rocca",
        "Eugen.Mouan",
        "Ferne.Dulling",
        "Gabrila.Gudyer",
        "Garnette.Brassington",
        "Gipsy.Swires",
        "Henrie.Beaze",
        "Inger.Landreth",
        "Jacki.McCathy",
        "Jon.Arnatt",
        "Kristofer.Ghiriardelli",
        "Leicester.Scamaden",
        "Lesly.Clubbe",
        "Lurlene.Daviot",
        "Maighdiln.Edwicker",
        "Maurizio.Humpage",
        "Miguela.Burril",
        "Minni.Maile",
        "Muhammad.Nevet",
        "Murielle.Stook",
        "Nicolea.Hevner",
        "Pearl.Lyes",
        "Peterus.Widmoor",
        "Phelia.Lapley",
```

Figure 1.3: List of usernames.

Retrieving the content of all fleets for the user with ID ”617398d4ce3cd7dfabac332c”,

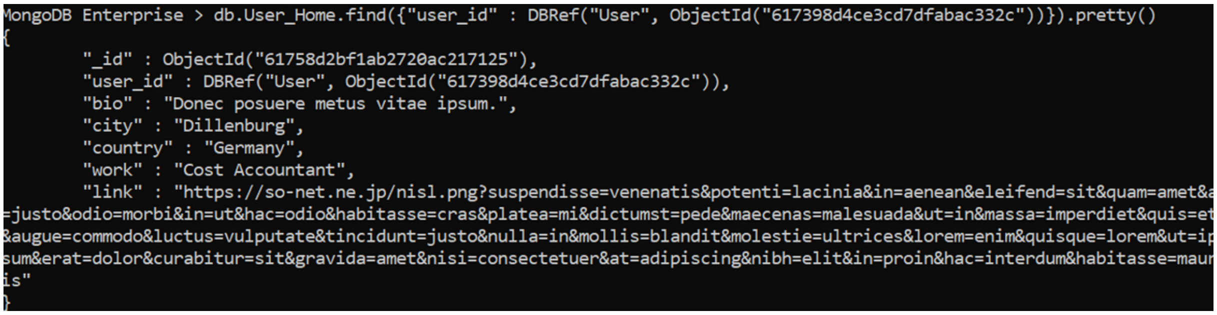


```
MongoDB Enterprise > db.Fleets.find({"username" : DBRef("User", ObjectId("617398d4ce3cd7dfabac332c"))}, {"fleet_content": 1}).pretty()
{
        "_id" : ObjectId("618a43a791e3d5b41c6c5638"),
        "fleet_content" : "http://dummyimage.com/138x100.png/cc0000/ffffff"
}
{

        "_id" : ObjectId("618a43a791e3d5b41c6c5639"),
        "fleet_content" : "http://dummyimage.com/207x100.png/5fa2dd/ffffff"
}
{

        "_id" : ObjectId("618a43a791e3d5b41c6c563a"),
        "fleet_content" : "http://dummyimage.com/247x100.png/cc0000/ffffff"
}
{

        "_id" : ObjectId("618a43a791e3d5b41c6c563b"),
        "fleet_content" : "http://dummyimage.com/217x100.png/5fa2dd/ffffff"
}
{

        "_id" : ObjectId("618a43a791e3d5b41c6c563c"),
        "fleet_content" : "http://dummyimage.com/102x100.png/ff4444/ffffff"
}
```

Figure 1.4: Fleets by user ”617398d4ce3cd7dfabac332c”.

For the user with ID ”617398d4ce3cd7dfabac332c”, retrieving the,

- First Name
- Last Name
- Username
- Email
- Date of Birth
- Age

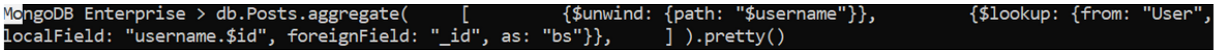


```
MongoDB Enterprise > db.User_Home.find({"user_id" : DBRef("User", ObjectId("617398d4ce3cd7dfabac332c"))}).pretty()
{
        "_id" : ObjectId("61758d2bf1ab2720ac217125"),
        "user_id" : DBRef("User", ObjectId("617398d4ce3cd7dfabac332c")),
        "bio" : "Donec posuere metus vitae ipsum.",
        "city" : "Dillenburg",
        "country" : "Germany",
        "work" : "Cost Accountant",
        "link" : "https://so-net.ne.jp/nisl.png?suspendisse=venenatis&potenti=lacinia&in=aenean&eleifend=sit&quam=amet&a
=justo&odio=morbi&in=ut&hac=odio&habitasse=cras&platea=mi&dictumst=pede&maecenas=malesuada&ut=in&massa=imperdiet&quis=et
&augue=commodo&luctus=vulputate&tincidunt=justo&nulla=in&mollis=blandit&molestie=ultrices&lorem=enim&quisque=lorem&ut=ip
sum&erat=dolor&curabitur=sit&gravida=amet&nisi=consectetuer&at=adipiscing&nibh=elit&in=proin&hac=interdum&habitasse=maur
is"
}
```

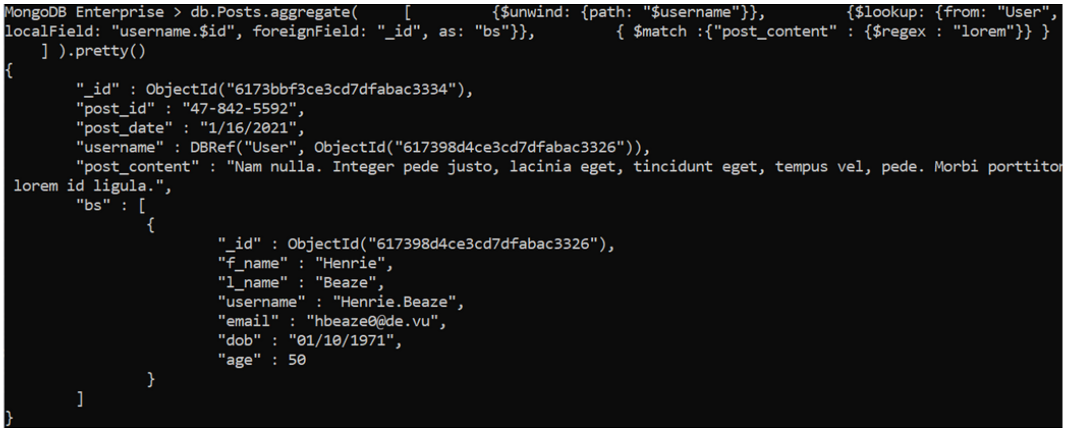
Figure 1.5: Details of user ”617398d4ce3cd7dfabac332c”.

1.2 Complex Queries

Retrieving all posts that contain the word "LOREM", and their corresponding user IDs,



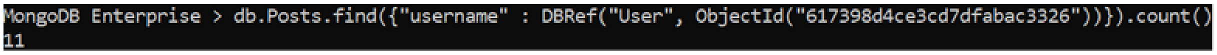
```
MongoDB Enterprise > db.Posts.aggregate(     [         {$unwind: {path: "$username"}},         {$lookup: {from: "User",
localField: "username.$id", foreignField: "_id", as: "bs"}},      ] ).pretty()
```



```
MongoDB Enterprise > db.Posts.aggregate(     [         {$unwind: {path: "$username"}},         {$lookup: {from: "User",
localField: "username.$id", foreignField: "_id", as: "bs"}},         { $match :{"post_content" : {$regex : "lorem"}} }
    ] ).pretty()
{
        "_id" : ObjectId("6173bbf3ce3cd7dfabac3334"),
        "post_id" : "47-842-5592",
        "post_date" : "1/16/2021",
        "username" : DBRef("User", ObjectId("617398d4ce3cd7dfabac3326")),
        "post_content" : "Nam nulla. Integer pede justo, lacinia eget, tincidunt eget, tempus vel, pede. Morbi porttito
 lorem id ligula.",
        "bs" : [
                {
                        "_id" : ObjectId("617398d4ce3cd7dfabac3326"),
                        "f_name" : "Henrie",
                        "l_name" : "Beaze",
                        "username" : "Henrie.Beaze",
                        "email" : "hbeaze0@de.vu",
                        "dob" : "01/10/1971",
                        "age" : 50
                }
        ]
}
```

Figure 1.6: Posts containing "LOREM".

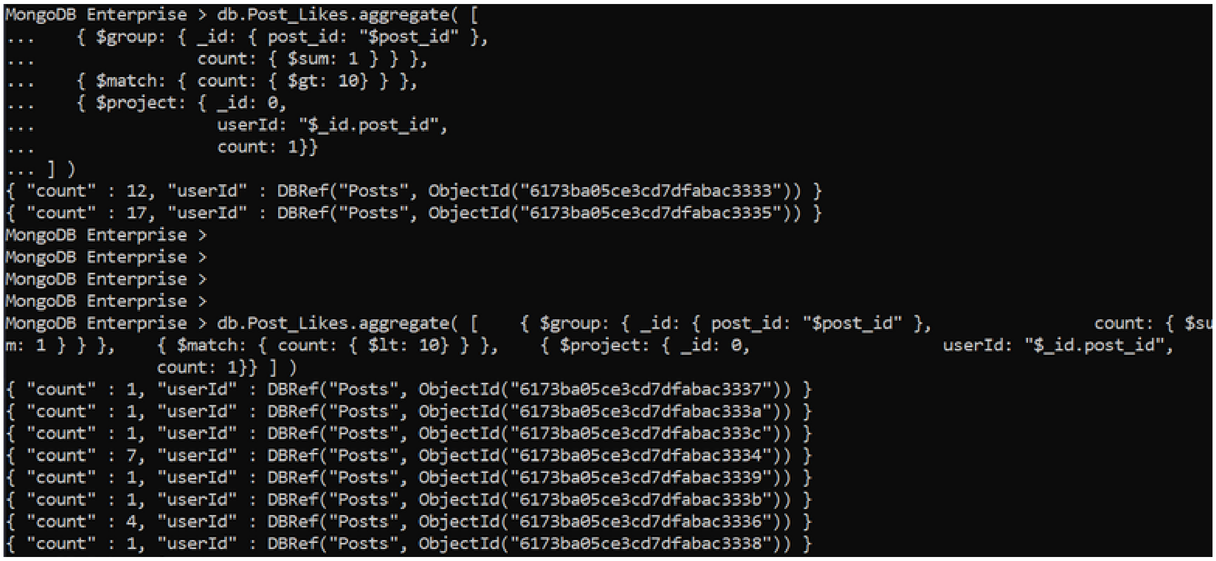
Retrieving the total number of posts for the user with ID "617398d4ce3cd7dfabac3326" via count,



```
MongoDB Enterprise > db.Posts.find({"username" : DBRef("User", ObjectId("617398d4ce3cd7dfabac3326"))}).count()
11
```

Figure 1.7: Posts by user "617398d4ce3cd7dfabac3326".

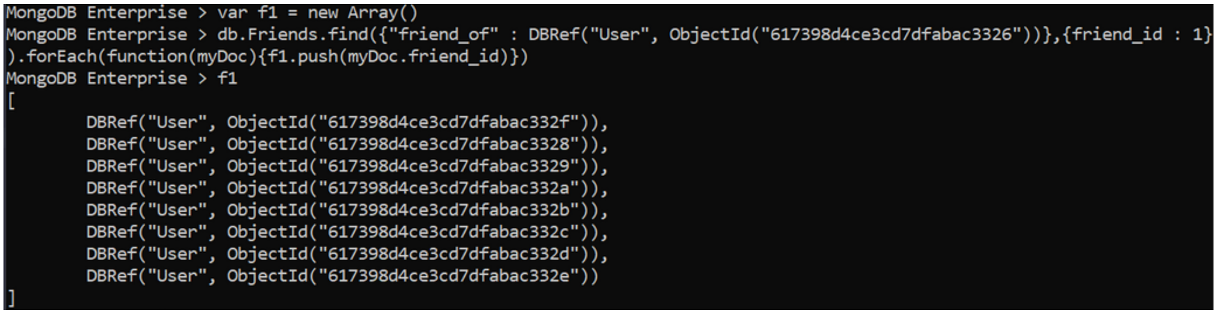
Retrieving all posts with over 10 likes and under 10 likes,



```
MongoDB Enterprise > db.Post_Likes.aggregate( [
...    { $group: { _id: { post_id: "$post_id" },
...                count: { $sum: 1 } } },
...    { $match: { count: { $gt: 10} } },
...    { $project: { _id: 0,
...                  userId: "$_id.post_id",
...                  count: 1}}
... ] )
{ "count" : 12, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac3333")) }
{ "count" : 17, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac3335")) }
MongoDB Enterprise >
MongoDB Enterprise >
MongoDB Enterprise >
MongoDB Enterprise >
MongoDB Enterprise > db.Post_Likes.aggregate( [    { $group: { _id: { post_id: "$post_id" },               count: { $su
m: 1 } } },    { $match: { count: { $lt: 10} } },    { $project: { _id: 0,                   userId: "$_id.post_id",
               count: 1}} ] )
{ "count" : 1, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac3337")) }
{ "count" : 1, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac333a")) }
{ "count" : 1, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac333c")) }
{ "count" : 7, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac3334")) }
{ "count" : 1, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac3339")) }
{ "count" : 1, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac333b")) }
{ "count" : 4, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac3336")) }
{ "count" : 1, "userId" : DBRef("Posts", ObjectId("6173ba05ce3cd7dfabac3338")) }
```

Figure 1.8: Posts with more or less than 10 likes.

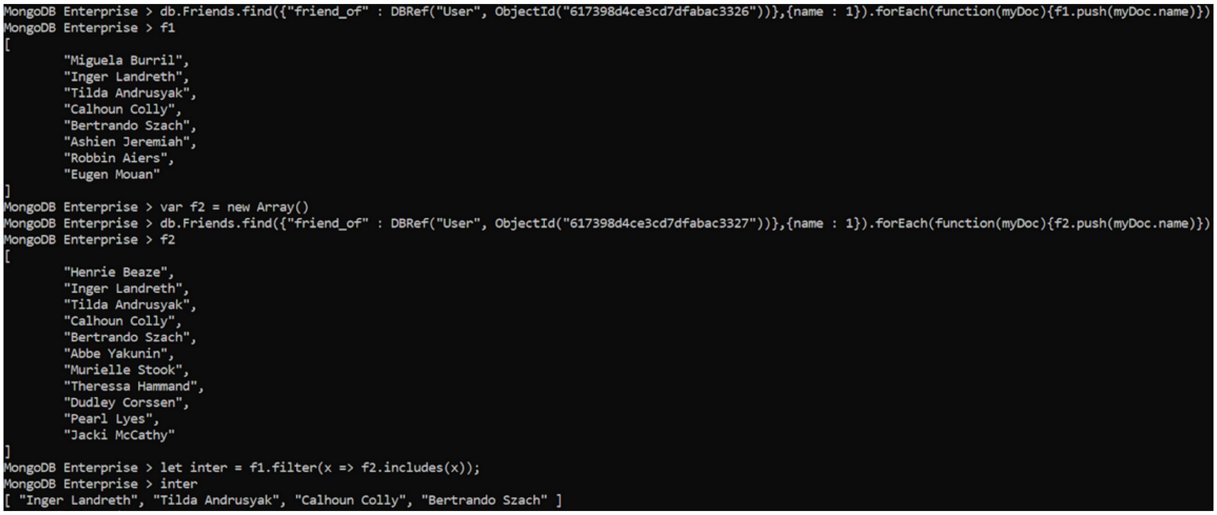
Retrieving the ID of all friends for the user with ID ”617398d4ce3cd7dfabac3326” and storing in an Array,



```
MongoDB Enterprise > var f1 = new Array()
MongoDB Enterprise > db.Friends.find({"friend_of" : DBRef("User", ObjectId("617398d4ce3cd7dfabac3326"))},{friend_id : 1}
).forEach(function(myDoc){f1.push(myDoc.friend_id)})
MongoDB Enterprise > f1
[
        DBRef("User", ObjectId("617398d4ce3cd7dfabac332f")),
        DBRef("User", ObjectId("617398d4ce3cd7dfabac3328")),
        DBRef("User", ObjectId("617398d4ce3cd7dfabac3329")),
        DBRef("User", ObjectId("617398d4ce3cd7dfabac332a")),
        DBRef("User", ObjectId("617398d4ce3cd7dfabac332b")),
        DBRef("User", ObjectId("617398d4ce3cd7dfabac332c")),
        DBRef("User", ObjectId("617398d4ce3cd7dfabac332d")),
        DBRef("User", ObjectId("617398d4ce3cd7dfabac332e"))
]
```

Figure 1.9: All friends of the user.

Retrieving all mutual friends between users with IDs ”617398d4ce3cd7dfabac3326” and ”617398d4ce3cd7dfabac3327” using filters,

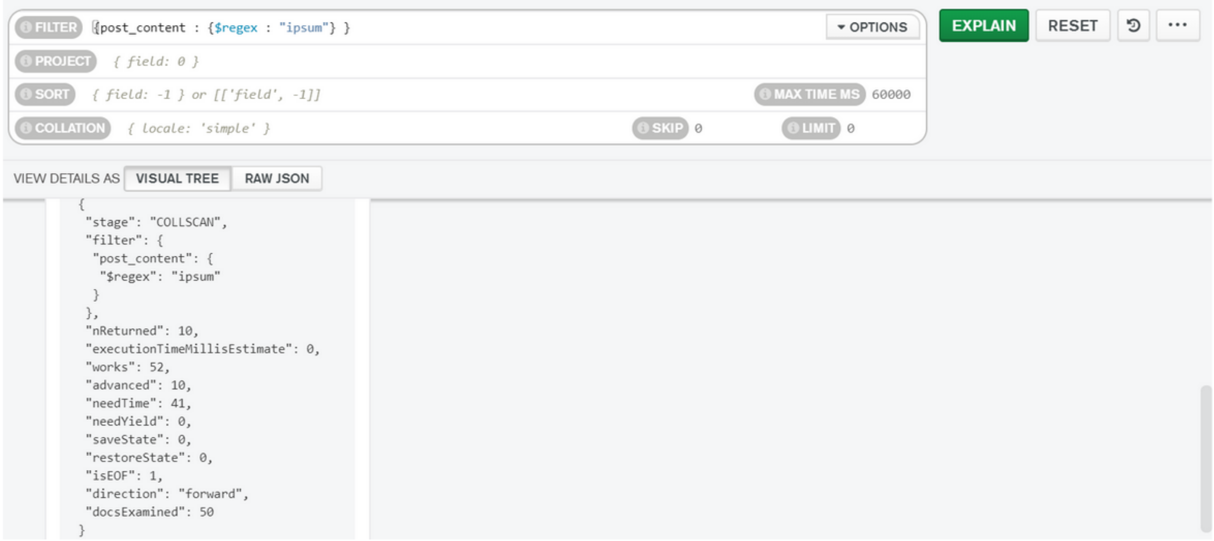


```
MongoDB Enterprise > db.Friends.find({"friend_of" : DBRef("User", ObjectId("617398d4ce3cd7dfabac3326"))},{name : 1}).forEach(function(myDoc){f1.push(myDoc.name)})
MongoDB Enterprise > f1
[
        "Miguela Burril",
        "Inger Landreth",
        "Tilda Andrusyak",
        "Calhoun Colly",
        "Bertrando Szach",
        "Ashien Jeremiah",
        "Robbin Aiers",
        "Eugen Mouan"
]
MongoDB Enterprise > var f2 = new Array()
MongoDB Enterprise > db.Friends.find({"friend_of" : DBRef("User", ObjectId("617398d4ce3cd7dfabac3327"))},{name : 1}).forEach(function(myDoc){f2.push(myDoc.name)})
MongoDB Enterprise > f2
[
        "Henrie Beaze",
        "Inger Landreth",
        "Tilda Andrusyak",
        "Calhoun Colly",
        "Bertrando Szach",
        "Abbe Yakunin",
        "Murielle Stook",
        "Theressa Hammand",
        "Dudley Corssen",
        "Pearl Lyes",
        "Jacki McCathy"
]
MongoDB Enterprise > let inter = f1.filter(x => f2.includes(x));
MongoDB Enterprise > inter
[ "Inger Landreth", "Tilda Andrusyak", "Calhoun Colly", "Bertrando Szach" ]
```

Figure 1.10: Common friends between the users.

1.3 Performance Analysis

Performance analysis on the query to retrieve all posts that contain the word ”IPSUM”, and their corresponding user IDs,



FILTER {post_content : {$regex : "ipsum"} }

MAX TIME MS 60000

SKIP 0

LIMIT 0

```
{
 "stage": "COLLSCAN",
 "filter": {
  "post_content": {
   "$regex": "ipsum"
  }
 },
 "nReturned": 10,
 "executionTimeMillisEstimate": 0,
 "works": 52,
 "advanced": 10,
 "needTime": 41,
 "needYield": 0,
 "saveState": 0,
 "restoreState": 0,
 "isEOF": 1,
 "direction": "forward",
 "docsExamined": 50
}
```

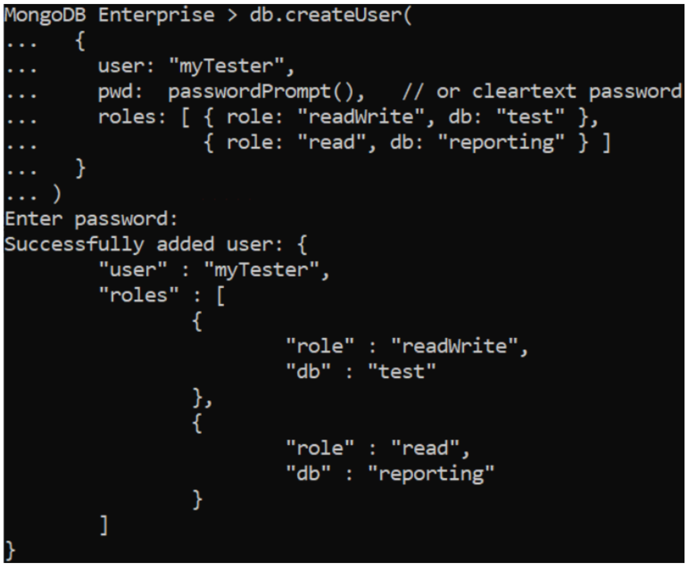
Figure 1.11: Performance analysis via MongoDB Compass.

Chapter 2

Multiple Users and Concurrency

2.1 Multiple Users and Privileges

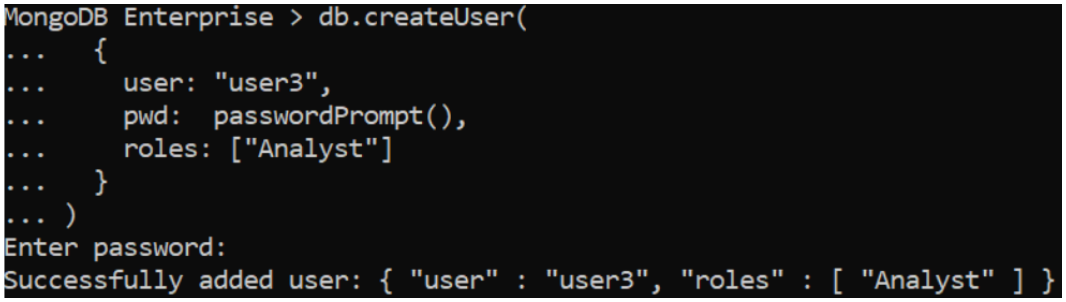
Creation of a user,



```
MongoDB Enterprise > db.createUser(
...   {
...     user: "myTester",
...     pwd:  passwordPrompt(),   // or cleartext password
...     roles: [ { role: "readWrite", db: "test" },
...              { role: "read", db: "reporting" } ]
...   }
... )
Enter password:
Successfully added user: {
        "user" : "myTester",
        "roles" : [
                {
                        "role" : "readWrite",
                        "db" : "test"
                },
                {
                        "role" : "read",
                        "db" : "reporting"
                }
        ]
}
```

Figure 2.1: Creating new user.

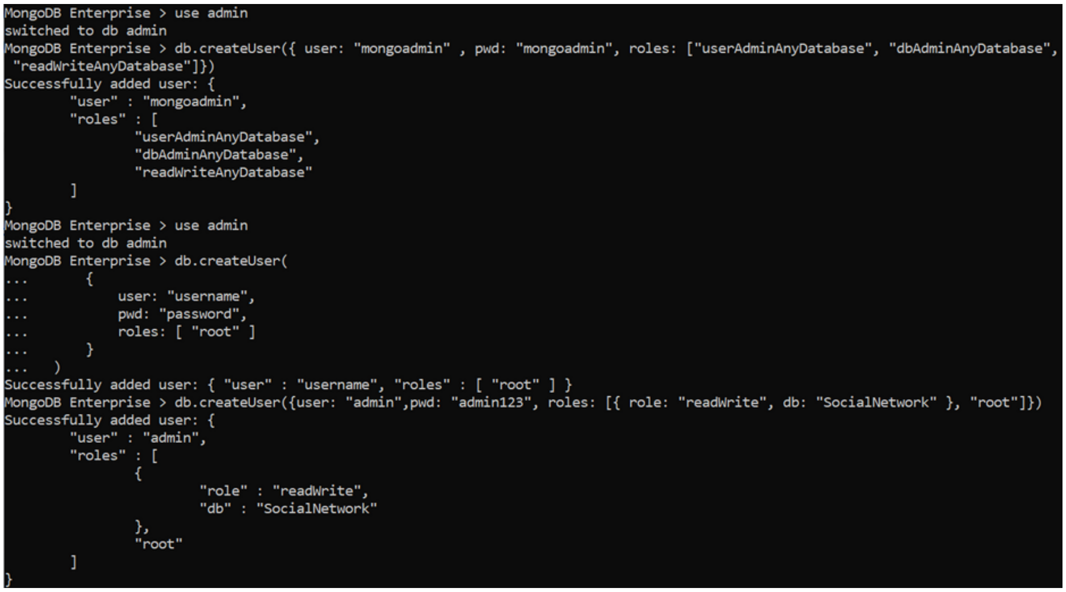
Creation of analyst role,



```
MongoDB Enterprise > db.createUser(
...   {
...     user: "user3",
...     pwd:  passwordPrompt(),
...     roles: ["Analyst"]
...   }
... )
Enter password:
Successfully added user: { "user" : "user3", "roles" : [ "Analyst" ] }
```

Figure 2.2: Creating analyst role.

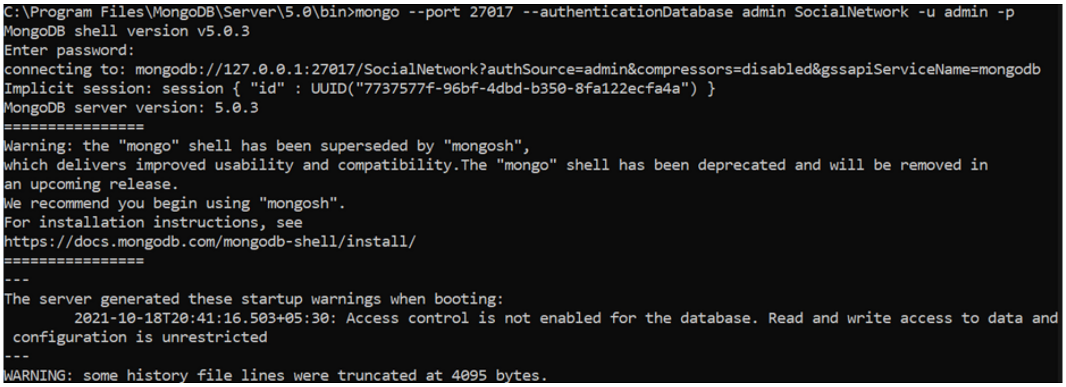
Creation of admin role,



```
MongoDB Enterprise > use admin
switched to db admin
MongoDB Enterprise > db.createUser({ user: "mongoadmin" , pwd: "mongoadmin", roles: ["userAdminAnyDatabase", "dbAdminAnyDatabase",
 "readWriteAnyDatabase"]})
Successfully added user: {
        "user" : "mongoadmin",
        "roles" : [
                "userAdminAnyDatabase",
                "dbAdminAnyDatabase",
                "readWriteAnyDatabase"
        ]
}
MongoDB Enterprise > use admin
switched to db admin
MongoDB Enterprise > db.createUser(
...       {
...           user: "username",
...           pwd: "password",
...           roles: [ "root" ]
...       }
...   )
Successfully added user: { "user" : "username", "roles" : [ "root" ] }
MongoDB Enterprise > db.createUser({user: "admin",pwd: "admin123", roles: [{ role: "readWrite", db: "SocialNetwork" }, "root"]})
Successfully added user: {
        "user" : "admin",
        "roles" : [
                {
                        "role" : "readWrite",
                        "db" : "SocialNetwork"
                },
                "root"
        ]
}
```

Figure 2.3: Creating admin role.

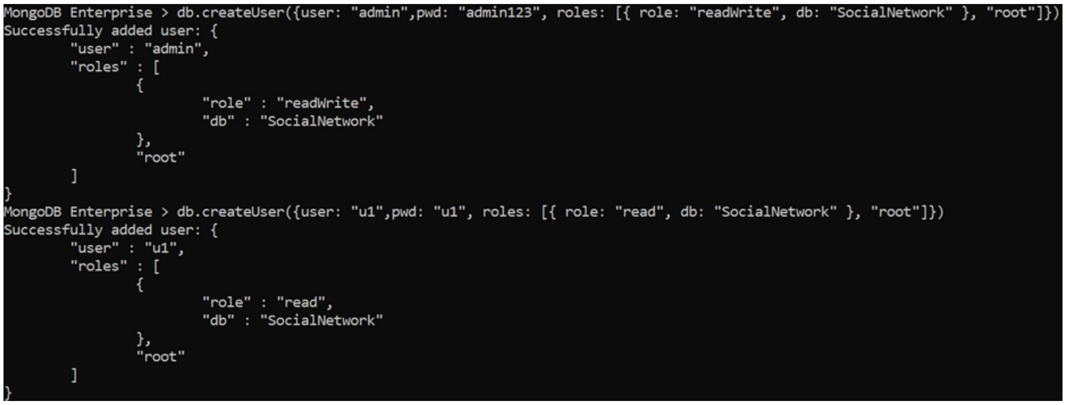
Connecting to admin,



```
C:\Program Files\MongoDB\Server\5.0\bin>mongo --port 27017 --authenticationDatabase admin SocialNetwork -u admin -p
MongoDB shell version v5.0.3
Enter password:
connecting to: mongodb://127.0.0.1:27017/SocialNetwork?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7737577f-96bf-4dbd-b350-8fa122ecfa4a") }
MongoDB server version: 5.0.3
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2021-10-18T20:41:16.503+05:30: Access control is not enabled for the database. Read and write access to data and
 configuration is unrestricted
---
WARNING: some history file lines were truncated at 4095 bytes.
```

Figure 2.4: Connecting to admin role.

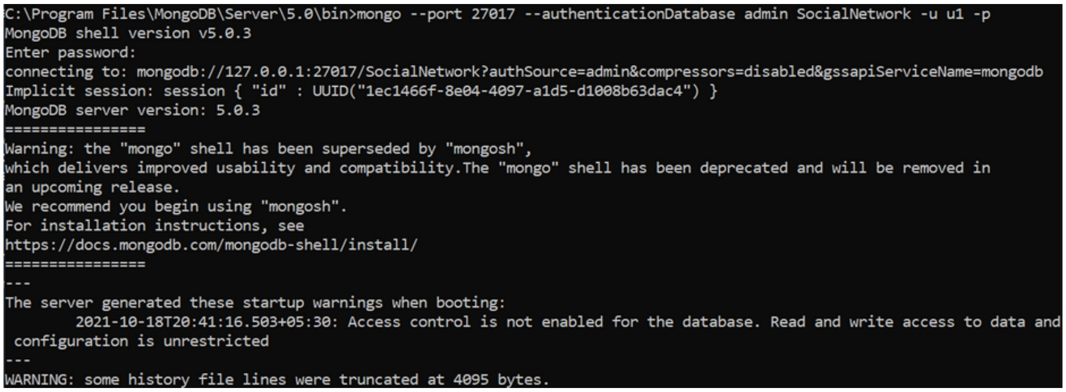
Creation of ”u1” user role,



```
MongoDB Enterprise > db.createUser({user: "admin",pwd: "admin123", roles: [{ role: "readWrite", db: "SocialNetwork" }, "root"]})
Successfully added user: {
        "user" : "admin",
        "roles" : [
                {
                        "role" : "readWrite",
                        "db" : "SocialNetwork"
                },
                "root"
        ]
}
MongoDB Enterprise > db.createUser({user: "u1",pwd: "u1", roles: [{ role: "read", db: "SocialNetwork" }, "root"]})
Successfully added user: {
        "user" : "u1",
        "roles" : [
                {
                        "role" : "read",
                        "db" : "SocialNetwork"
                },
                "root"
        ]
}
```

Figure 2.5: Creating user role.

Connecting to u1,

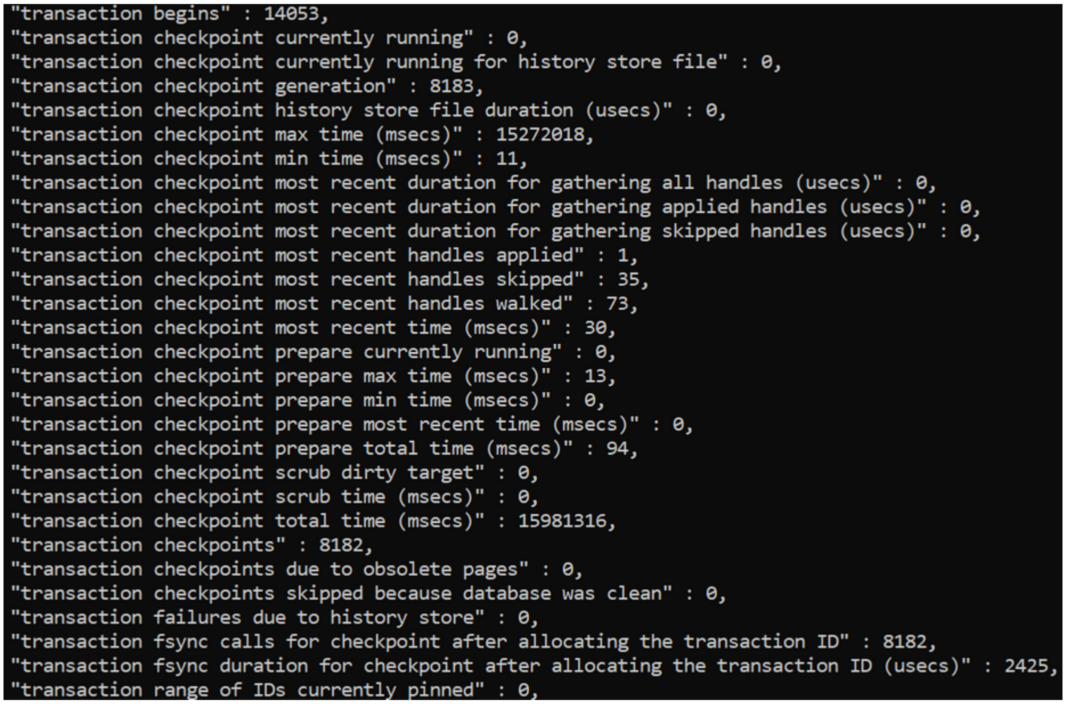


```
C:\Program Files\MongoDB\Server\5.0\bin>mongo --port 27017 --authenticationDatabase admin SocialNetwork -u u1 -p
MongoDB shell version v5.0.3
Enter password:
connecting to: mongodb://127.0.0.1:27017/SocialNetwork?authSource=admin&compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("1ec1466f-8e04-4097-a1d5-d1008b63dac4") }
MongoDB server version: 5.0.3
================
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility.The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
================
---
The server generated these startup warnings when booting:
        2021-10-18T20:41:16.503+05:30: Access control is not enabled for the database. Read and write access to data and
 configuration is unrestricted
---
WARNING: some history file lines were truncated at 4095 bytes.
```

Figure 2.6: Connecting to user role.

2.2 Concurrent Transactions

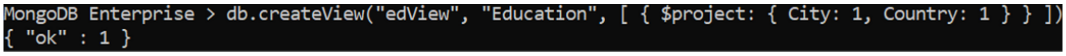
Initial list of transactions via db.serverStatus(),



```
"transaction begins" : 14053,
"transaction checkpoint currently running" : 0,
"transaction checkpoint currently running for history store file" : 0,
"transaction checkpoint generation" : 8183,
"transaction checkpoint history store file duration (usecs)" : 0,
"transaction checkpoint max time (msecs)" : 15272018,
"transaction checkpoint min time (msecs)" : 11,
"transaction checkpoint most recent duration for gathering all handles (usecs)" : 0,
"transaction checkpoint most recent duration for gathering applied handles (usecs)" : 0,
"transaction checkpoint most recent duration for gathering skipped handles (usecs)" : 0,
"transaction checkpoint most recent handles applied" : 1,
"transaction checkpoint most recent handles skipped" : 35,
"transaction checkpoint most recent handles walked" : 73,
"transaction checkpoint most recent time (msecs)" : 30,
"transaction checkpoint prepare currently running" : 0,
"transaction checkpoint prepare max time (msecs)" : 13,
"transaction checkpoint prepare min time (msecs)" : 0,
"transaction checkpoint prepare most recent time (msecs)" : 0,
"transaction checkpoint prepare total time (msecs)" : 94,
"transaction checkpoint scrub dirty target" : 0,
"transaction checkpoint scrub time (msecs)" : 0,
"transaction checkpoint total time (msecs)" : 15981316,
"transaction checkpoints" : 8182,
"transaction checkpoints due to obsolete pages" : 0,
"transaction checkpoints skipped because database was clean" : 0,
"transaction failures due to history store" : 0,
"transaction fsync calls for checkpoint after allocating the transaction ID" : 8182,
"transaction fsync duration for checkpoint after allocating the transaction ID (usecs)" : 2425,
"transaction range of IDs currently pinned" : 0,
```

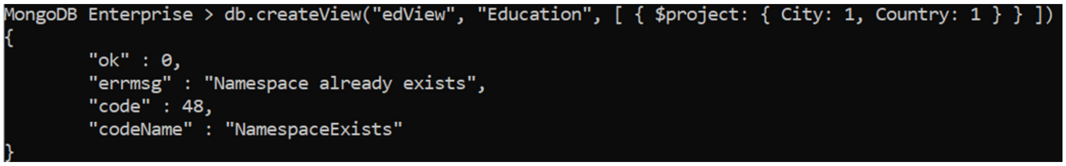
Figure 2.7: Initial transaction list.

Creation of conflicting view between ”admin” and ”u1”,



```
MongoDB Enterprise > db.createView("edView", "Education", [ { $project: { City: 1, Country: 1 } } ])
{ "ok" : 1 }
```

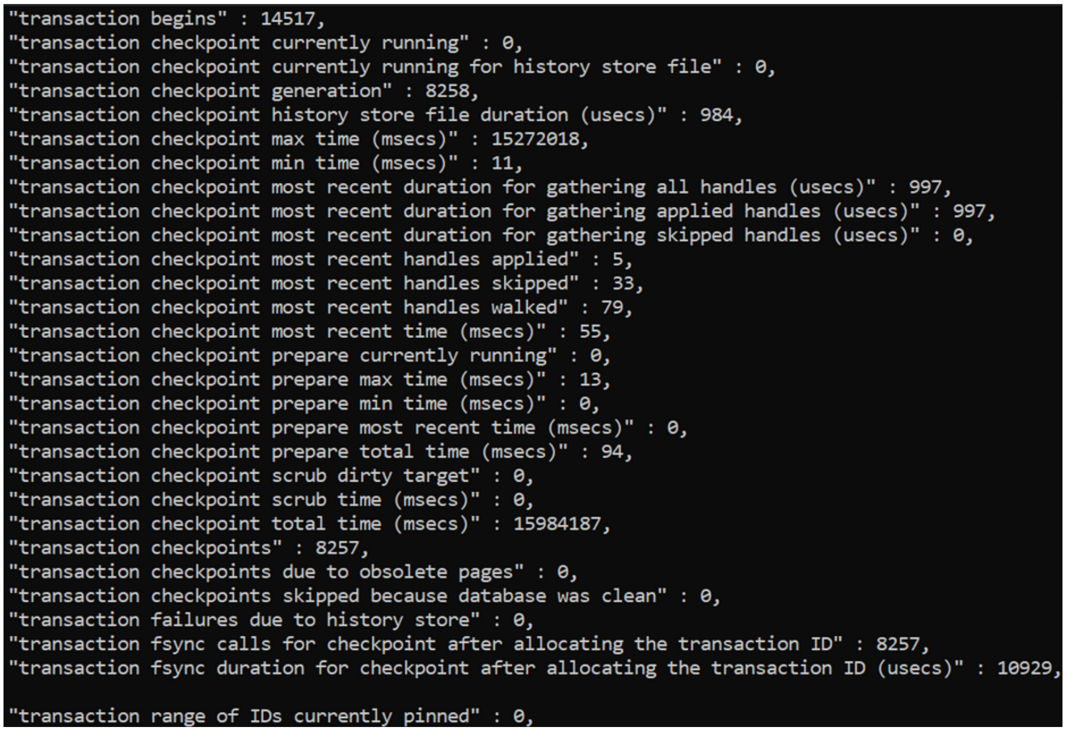
Figure 2.8: View in ”admin”.



```
MongoDB Enterprise > db.createView("edView", "Education", [ { $project: { City: 1, Country: 1 } } ])
{
        "ok" : 0,
        "errmsg" : "Namespace already exists",
        "code" : 48,
        "codeName" : "NamespaceExists"
}
```

Figure 2.9: View in ”u1”.

Updated list of transactions via db.serverStatus() after creation of a conflicting view between ”admin” and ”u1”,



```
"transaction begins" : 14517,
"transaction checkpoint currently running" : 0,
"transaction checkpoint currently running for history store file" : 0,
"transaction checkpoint generation" : 8258,
"transaction checkpoint history store file duration (usecs)" : 984,
"transaction checkpoint max time (msecs)" : 15272018,
"transaction checkpoint min time (msecs)" : 11,
"transaction checkpoint most recent duration for gathering all handles (usecs)" : 997,
"transaction checkpoint most recent duration for gathering applied handles (usecs)" : 997,
"transaction checkpoint most recent duration for gathering skipped handles (usecs)" : 0,
"transaction checkpoint most recent handles applied" : 5,
"transaction checkpoint most recent handles skipped" : 33,
"transaction checkpoint most recent handles walked" : 79,
"transaction checkpoint most recent time (msecs)" : 55,
"transaction checkpoint prepare currently running" : 0,
"transaction checkpoint prepare max time (msecs)" : 13,
"transaction checkpoint prepare min time (msecs)" : 0,
"transaction checkpoint prepare most recent time (msecs)" : 0,
"transaction checkpoint prepare total time (msecs)" : 94,
"transaction checkpoint scrub dirty target" : 0,
"transaction checkpoint scrub time (msecs)" : 0,
"transaction checkpoint total time (msecs)" : 15984187,
"transaction checkpoints" : 8257,
"transaction checkpoints due to obsolete pages" : 0,
"transaction checkpoints skipped because database was clean" : 0,
"transaction failures due to history store" : 0,
"transaction fsync calls for checkpoint after allocating the transaction ID" : 8257,
"transaction fsync duration for checkpoint after allocating the transaction ID (usecs)" : 10929,

"transaction range of IDs currently pinned" : 0,
```

Figure 2.10: Updated transaction list.

Chapter 3

Contributions

1. Kartika Nair - PES1UG19CS213
   - Complete Working Model of Database Application, Multiple Users and Concurrency, Report
   - Hours Spent - 5
2. Krithika Ragothaman - PES1UG19CS231
   - Complete Working Model of Database Application, Multiple Users and Concurrency
   - Hours Spent - 5
3. Maitreyi P - PES1UG19CS254
   - Complete Working Model of Database Application, Multiple Users and Concurrency
   - Hours Spent - 5