

Experiment report: Document DataBases

Done by student: **Bylkova Kristina (伯汀娜)**, 1820249049

Course: **Big Data Analysis Technology**

Date: **September 2024**

Summary

Document databases are a type of NoSQL database that store data in the form of documents. Unlike traditional relational databases that store data in rows and columns, document databases use a flexible, schema-less model, allowing data to be stored in structures such as JSON, BSON, or XML. Each document can have a unique structure, making document databases ideal for handling unstructured or semi-structured data.

This report shows the MindMap of Document DB and the use of the Mongo DataBase. MongoDB is a document database designed for ease of application development and scaling. Its main key features are horizontal scalability, high availability, flexibility and high performance with rich query mechanism.

MindMap

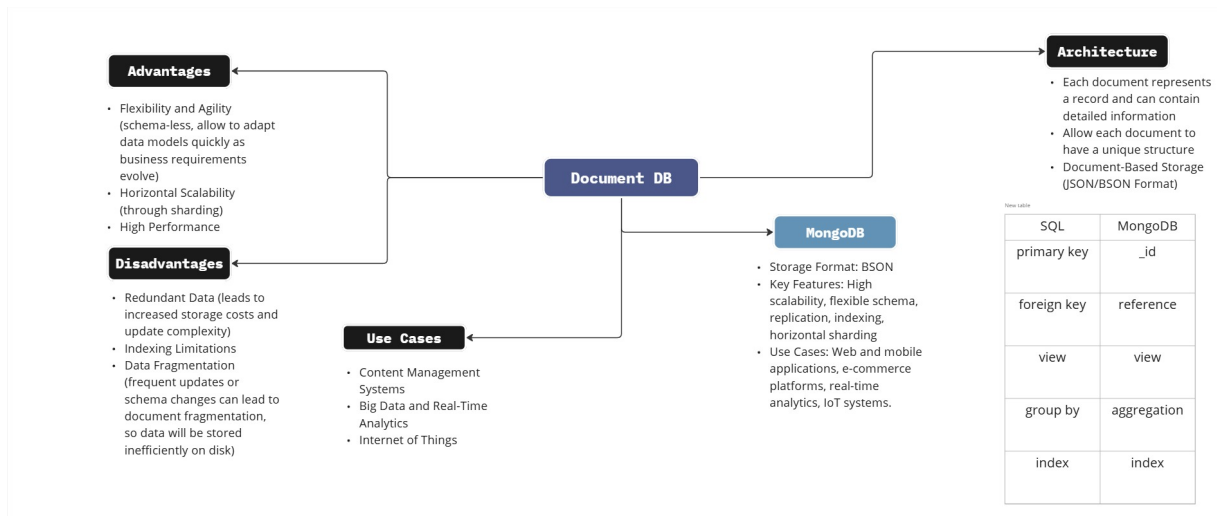
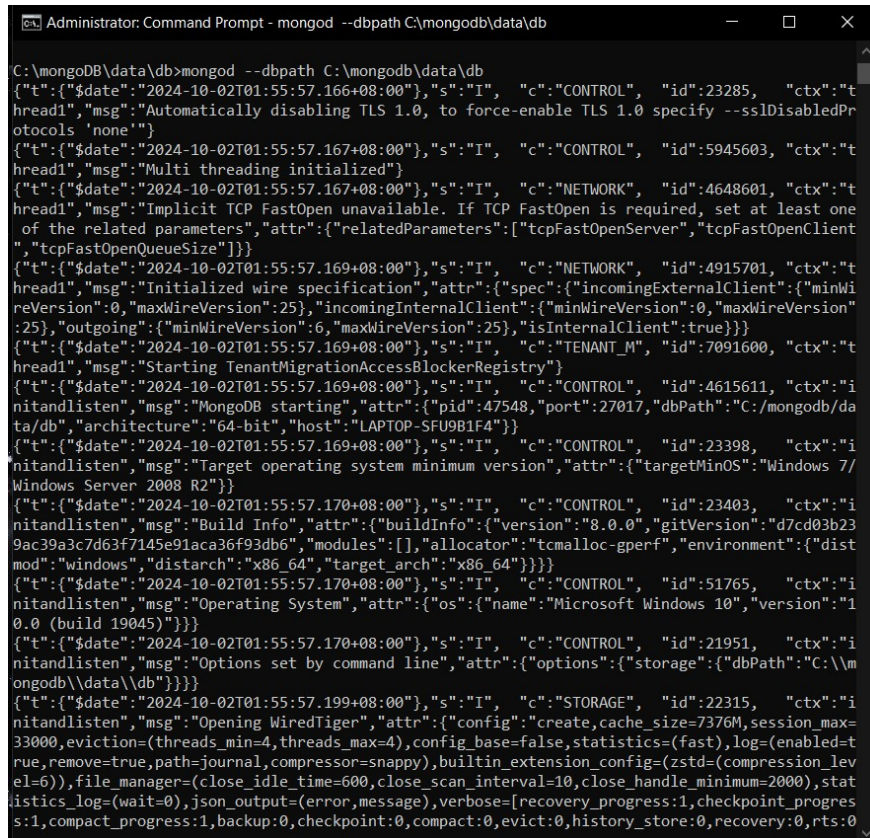


Figure 1: MindMap for DocumentDB

Installation

I installed and configured MongoDB and Compass via the guide from the team that performed.



```
Administrator: Command Prompt - mongod --dbpath C:\mongodb\data\db

C:\mongoDB\data\db>mongod --dbpath C:\mongodb\data\db
{"t":{"date":"2024-10-02T01:55:57.166+08:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":"t
hread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledPr
otocols 'none'"}
{"t":{"date":"2024-10-02T01:55:57.167+08:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":"t
hread1","msg":"Multi threading initialized"}
{"t":{"date":"2024-10-02T01:55:57.167+08:00"},"s":"I", "c":"NETWORK", "id":4648601, "ctx":"t
hread1","msg":"Implicit TCP FastOpen unavailable. If TCP FastOpen is required, set at least one
of the related parameters","attr":{"relatedParameters":["tcpFastOpenServer","tcpFastOpenClient
","tcpFastOpenQueueSize"]}}
{"t":{"date":"2024-10-02T01:55:57.169+08:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":"t
hread1","msg":"Initialized wire specification","attr":{"spec":{"incomingExternalClient":{"minWi
reVersion":0,"maxWireVersion":25},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion
":25},"outgoing":{"minWireVersion":6,"maxWireVersion":25},"isInternalClient":true}}}}
{"t":{"date":"2024-10-02T01:55:57.169+08:00"},"s":"I", "c":"TENANT_M", "id":7091600, "ctx":"t
hread1","msg":"Starting TenantMigrationAccessBlockerRegistry"}
{"t":{"date":"2024-10-02T01:55:57.169+08:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":"i
nitandlisten","msg":"MongoDB starting","attr":{"pid":47548,"port":27017,"dbPath":"C:/mongodb/da
ta/db","architecture":"64-bit","host":"LAPTOP-SFU9B1F4"}}
{"t":{"date":"2024-10-02T01:55:57.169+08:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":"i
nitandlisten","msg":"Target operating system minimum version","attr":{"targetMinOS":"Windows 7/
Windows Server 2008 R2"}}
{"t":{"date":"2024-10-02T01:55:57.170+08:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":"i
nitandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"8.0.0","gitVersion":"d7cd03b23
9ac39a3c7d63f7145e91aca36f93db6","modules":[],"allocator":"tcmalloc-gperf","environment":{"dist
mod":"windows","distarch":"x86_64","target_arch":"x86_64"}}}}
{"t":{"date":"2024-10-02T01:55:57.170+08:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":"i
nitandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Windows 10","version":"1
0.0 (build 19045)}}}}
{"t":{"date":"2024-10-02T01:55:57.170+08:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":"i
nitandlisten","msg":"Options set by command line","attr":{"options":{"storage":{"dbPath":"C:\\m
ongodb\\data\\db"}}}}
{"t":{"date":"2024-10-02T01:55:57.199+08:00"},"s":"I", "c":"STORAGE", "id":22315, "ctx":"i
nitandlisten","msg":"Opening WiredTiger","attr":{"config":"create,cache_size=7376M,session_max=
33000,eviction=(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log=(enabled=t
rue,remove=true,path=journal,compressor=snappy),builtin_extension_config=(zstd=(compression_lev
el=6)),file_manager=(close_idle_time=600,close_scan_interval=10,close_handle_minimum=2000),stat
istics_log=(wait=0),json_output=(error,message),verbose=[recovery_progress:1,checkpoint_progres
s:1,compact_progress:1,backup:0,checkpoint:0,compact:0,evict:0,history_store:0,recovery:0,rts:0
```

Figure 2: Starting the MongoDB service

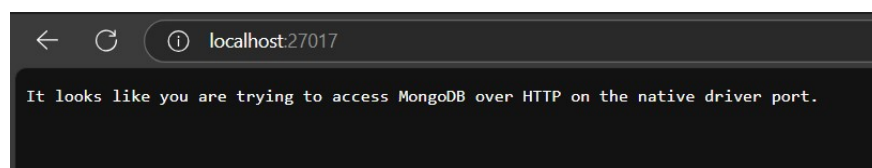


Figure 3: Result of starting

Also I configured logs for starting MongoDB.

```
C:\mongoDB\bin>mongod dbpath "C:\mongoDB\data\db" -logpath "C:\mongoDB\data\log" -install -serviceName "MongoDB"
{"t":{"$date":"2024-10-01T18:26:47.728Z"},"s":"F", "c":"CONTROL", "id":20574, "ctx":"thread
1","msg":"Error during global initialization","attr":{"error":{"code":38,"codeName":"FileNotOpe
n"},"errmsg":"logpath \"C:\\mongoDB\\data\\log\" should name a file, not a directory."}}
C:\mongoDB\bin>net start MongoDB
The MongoDB Server (MongoDB) service is starting.
The MongoDB Server (MongoDB) service was started successfully.
```

Figure 4: Start via net start MongoDB

And connected via Compass.

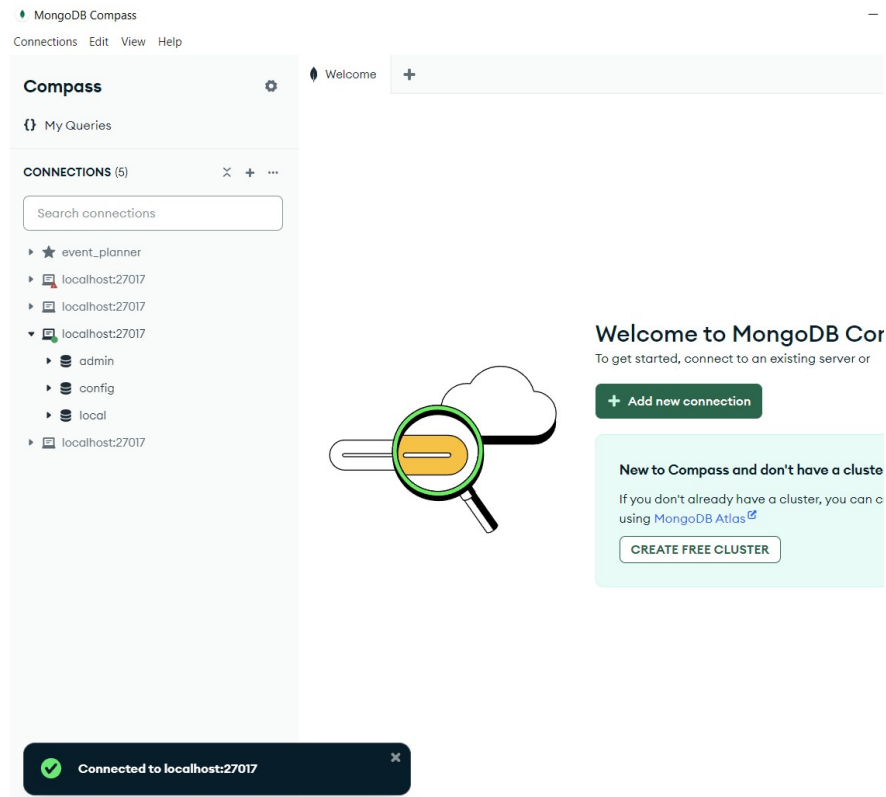


Figure 5: Connected in Compass

CRUD operations

I used basic commands in MongoDB shell here.

```
>_MONGOSH
> db.createCollection("sessions")
< { ok: 1 }
> db.sessions.insertOne({ "user": "Alice", "createdAt": new Date() })
< {
  acknowledged: true,
  insertedId: ObjectId('66fd0661fc5f7dd48be6943d')
}
> db.sessions.find().pretty()
< {
  _id: ObjectId('66fd0661fc5f7dd48be6943d'),
  user: 'Alice',
  createdAt: 2024-10-02T08:37:53.993Z
}
test>
```

Figure 6: Create and Insert One

```
>_MONGOSH
> db.sessions.deleteOne({ "user": "Alice Smith" })
< {
  acknowledged: true,
  deletedCount: 1
}
> db.sessions.find().pretty()
< {
  _id: ObjectId('66fd068bee1e1f5e519f2501'),
  user: 'Bob',
  createdAt: 2024-10-02T08:38:35.810Z,
  status: 'active'
}
{
  _id: ObjectId('66fd068bee1e1f5e519f2502'),
  user: 'Charlie',
  createdAt: 2024-10-02T08:38:35.810Z
}
test>
```

Figure 7: Delete One

```

>_MONGOSH
> db.sessions.insertMany([
  { "user": "Bob", "createdAt": new Date() },
  { "user": "Charlie", "createdAt": new Date() }
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66fd068bee1f5e519f2501'),
    '1': ObjectId('66fd068bee1f5e519f2502')
  }
}
> db.sessions.find().pretty()
< [
  {
    _id: ObjectId('66fd0661fc5f7dd48be6943d'),
    user: 'Alice',
    createdAt: 2024-10-02T08:37:53.993Z
  },
  {
    _id: ObjectId('66fd068bee1f5e519f2501'),
    user: 'Bob',
    createdAt: 2024-10-02T08:38:35.810Z
  },
  {
    _id: ObjectId('66fd068bee1f5e519f2502'),
    user: 'Charlie',
    createdAt: 2024-10-02T08:38:35.810Z
  }
]
test>|

```

Figure 8: Insert Many

```

>_MONGOSH
> db.sessions.deleteMany({})
< {
  acknowledged: true,
  deletedCount: 2
}
> db.sessions.find().pretty()
<
test>|

```

Figure 9: Delete Many

```

>_MONGOSH

> db.sessions.updateOne(
  { "user": "Alice" }, // Query condition
  { $set: { "user": "Alice Smith" } } // Update content
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.sessions.find().pretty()
< {
  _id: ObjectId('66fd0661fc5f7dd48be6943d'),
  user: 'Alice Smith',
  createdAt: 2024-10-02T08:37:53.993Z
}
{
  _id: ObjectId('66fd068bee1e1f5e519f2501'),
  user: 'Bob',
  createdAt: 2024-10-02T08:38:35.810Z
}
{
  _id: ObjectId('66fd068bee1e1f5e519f2502'),
  user: 'Charlie',
  createdAt: 2024-10-02T08:38:35.810Z
}
test>

```

Figure 10: Update One

Code

Test 1: Generate and insert data

```
1 import random
2 from datetime import datetime, timedelta
3 from faker import Faker
4 from pymongo import MongoClient
5
6 fake = Faker()
7 fake.seed_instance(42)
8 random.seed(42)
9 # MongoDB connection settings
10 mongo_client = MongoClient('mongodb://localhost:27017/')
11 db = mongo_client['test5']
12 collection = db['test5']
13 collection.delete_many({})
14
15
16 def generate_data(num_records):
17     categories = ['Category A', 'Category B', 'Category C', 'Category D']
18     tags = ["tag1", "tag2", "tag3", "tag4", "tag5"]
19     data = []
20
21     for _ in range(num_records):
22         record = {
23             "_id": fake.uuid4(),
24             "name": fake.name(),
25             "description": fake.text(max_nb_chars=200),
26             "category": random.choice(categories),
27             "price": round(random.uniform(10.0, 500.0), 2),
28             "location": {
29                 "type": "Point",
30                 "coordinates": [
31                     round(random.uniform(-180, 180), 6), # Longitude
32                     round(random.uniform(-90, 90), 6) # Latitude
33                 ]
34             },
35             "tags": [random.choice(tags) for _ in range(random.randint(1, 5))],
36             "ratings": [
37                 {
38                     "userId": random.randint(100, 1000),
39                     "rating": round(random.uniform(1.0, 5.0), 1),
40                     "timestamp": datetime.strftime(
41                         datetime.utcnow() - timedelta(days=random.randint(0, 365)),
42                         '%Y-%m-%dT%H:%M:%SZ'
43                     )
44                 } for _ in range(random.randint(1, 5))
45             ]
46         }
47         # data.append(record)
48         # Bulk insert data
49         collection.insert_one(record)
```

```

50         if _ % 1000 == 0:
51             print(f"Inserted {_} data.")
52
53
54 # Generate 100,000 records and insert them into MongoDB
55 num_records = 100000
56 data_set = generate_data(num_records)
57
58 print(f"Successfully inserted {num_records} data into MongoDB.")

```

Test 2: Aggregation

```

1 import random
2 from datetime import datetime, timedelta
3 from faker import Faker
4 from pymongo import MongoClient
5
6 fake = Faker()
7 fake.seed_instance(42)
8 random.seed(42)
9 # MongoDB connection settings
10 mongo_client = MongoClient('mongodb://localhost:27017/')
11 db = mongo_client['test5']
12 collection = db['test5']
13
14 pipeline = [
15     {
16         "$group": {
17             "_id": "$category",
18             "count": {"$sum": 1}
19         }
20     }
21 ]
22
23 result = list(collection.aggregate(pipeline))
24 for item in result:
25     print(f"Category: {item['_id']}, Count: {item['count']}")

```

Test 3: Calculate Average Price per Category

```

1 import random
2 from datetime import datetime, timedelta
3 from faker import Faker
4 from pymongo import MongoClient
5
6 fake = Faker()
7 fake.seed_instance(42)
8 random.seed(42)
9 # MongoDB connection settings
10 mongo_client = MongoClient('mongodb://localhost:27017/')
11 db = mongo_client['test5']
12 collection = db['test5']
13
14 pipeline = [

```



```

15     {
16         "$group": {
17             "_id": "$category",
18             "average_price": {"$avg": "$price"}
19         }
20     }
21 ]
22
23 result = list(collection.aggregate(pipeline))
24 for item in result:
25     print(f"Category: {item['_id']}, Average Price: {item['average_price']:.2f}")

```

Test 4: Measure the time with no index

```

1 import random
2 from datetime import datetime, timedelta
3 from faker import Faker
4 from pymongo import MongoClient
5
6 fake = Faker()
7 fake.seed_instance(42)
8 random.seed(42)
9 # MongoDB connection settings
10 mongo_client = MongoClient('mongodb://localhost:27017/')
11 db = mongo_client['test5']
12 collection = db['test5']
13
14 search_point = {
15     "type": "Point",
16     "coordinates": [-50.781527, -32.869214]
17 }
18 near_query = {
19     "location": {
20         "$geoWithin": {
21             "$centerSphere": [search_point['coordinates'], 1000 / 6378.1]
22         }
23     }
24 }
25 import time
26 start_time = time.time()
27 cursor = collection.find(near_query)
28 for doc in cursor:
29     print(doc)
30 end_time = time.time()
31 print(f"measure the time with no index: {end_time - start_time:.4f}s")
32
33 collection.create_index([('location', '2dsphere')])

```

Test 5: Create index of text and measure the time after creating index

```

1 import random
2 from datetime import datetime, timedelta
3 from faker import Faker
4 from pymongo import MongoClient

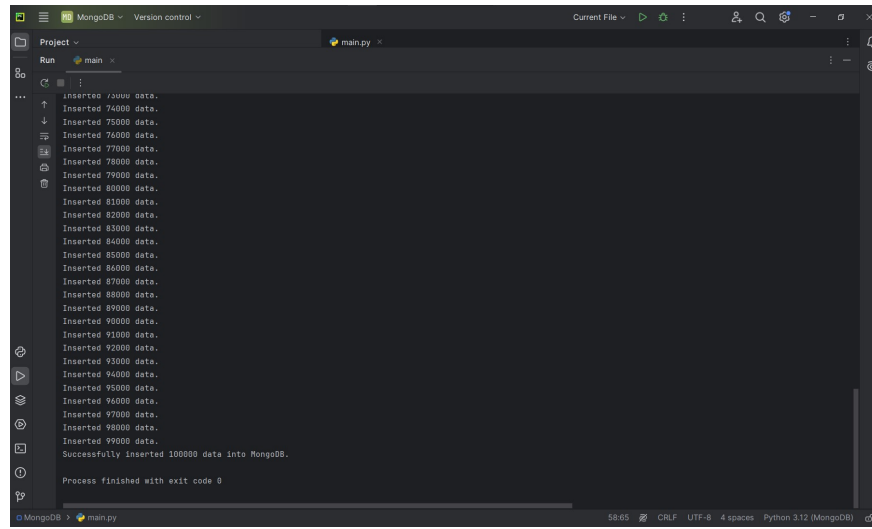
```

```

5
6 fake = Faker()
7 fake.seed_instance(42)
8 random.seed(42)
9 # MongoDB connection settings
10 mongo_client = MongoClient('mongodb://localhost:27017/')
11 db = mongo_client['test5']
12 collection = db['test5']
13
14 # create index of text
15 collection.create_index([('description', 'text')])
16
17 # measure the time after creating index
18 import time
19 start_time = time.time()
20 search_query = {'$text': {'$search': 'administration'}}
21 cursor = collection.find(search_query)
22 for doc in cursor:
23     print(doc)
24 end_time = time.time()
25 print(f"measure the time after creating index: {end_time - start_time:.4f}s")
26
27 print("current Index: ")
28 for index_info in collection.list_indexes():
29     print(index_info)

```

Code Execution



```
Project ▾ MongoDB ▾ Version control ▾
Run main
main.py
...
↑
↓
Inserted 74000 data.
Inserted 75000 data.
Inserted 76000 data.
Inserted 77000 data.
Inserted 78000 data.
Inserted 79000 data.
Inserted 80000 data.
Inserted 81000 data.
Inserted 82000 data.
Inserted 83000 data.
Inserted 84000 data.
Inserted 85000 data.
Inserted 86000 data.
Inserted 87000 data.
Inserted 88000 data.
Inserted 89000 data.
Inserted 90000 data.
Inserted 91000 data.
Inserted 92000 data.
Inserted 93000 data.
Inserted 94000 data.
Inserted 95000 data.
Inserted 96000 data.
Inserted 97000 data.
Inserted 98000 data.
Inserted 99000 data.
Successfully Inserted 100000 data into MongoDB.
Process finished with exit code 0
MongoDB main.py 58:05 CRLF UTF-8 4 spaces Python 3.12 (MongoDB)
```

Figure 11: Test 1

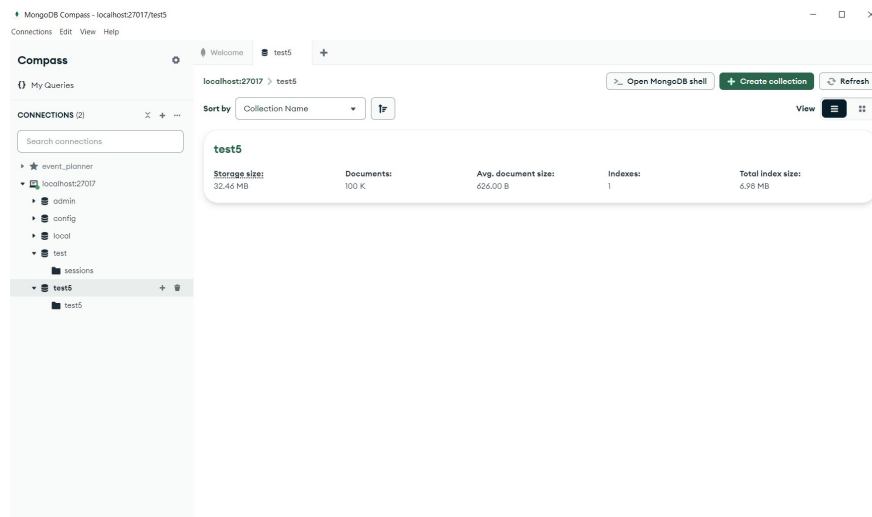


Figure 12: Test 1 (Compass)

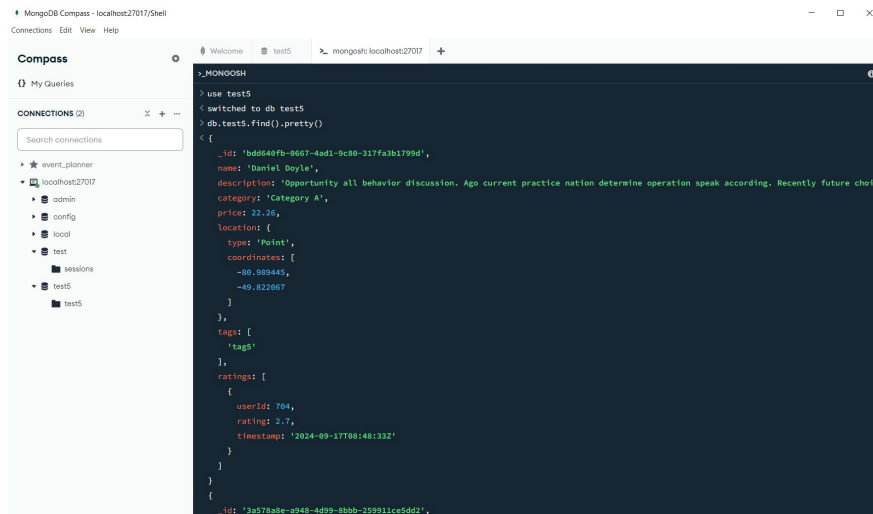


Figure 13: Test 1 (MongoDB Shell)

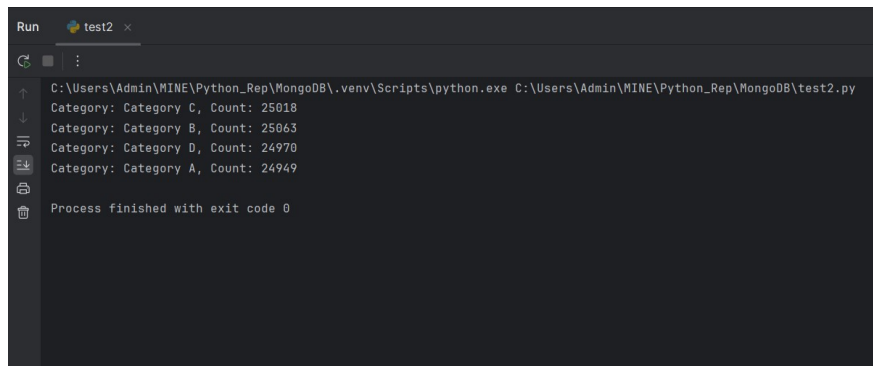


Figure 14: Test 2

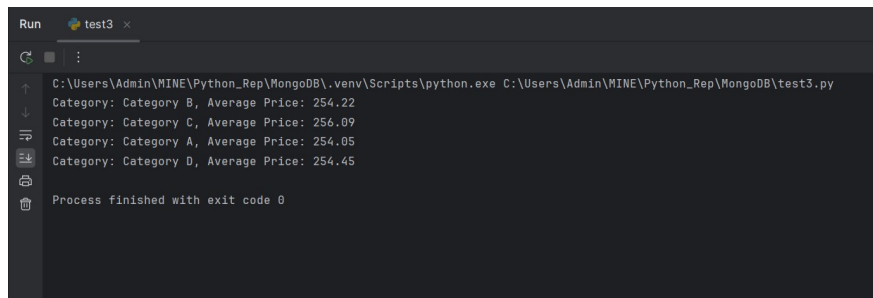


Figure 15: Test 3

```
Run test4
{
  "_id": "514422de-e4d0-4dd5-8c33-b55bhebla885d", "name": "Alan Mercer", "description": "Exactly situation discuss they. Artist sort answer list six word strong.\nSci",
  "_id": "ea41d0ce-f883-437b-bf62-bb0c1cd49634", "name": "Natalie Leonard", "description": "That perhaps culture small. Identify hand account simple difficult bar. He",
  "_id": "ea0edeb5a-850f-4f8c-bcdc-5d81c95b40ba", "name": "Kendra Levy", "description": "True five morning thing.\nTown clearly society sea discover Republican. Themse",
  "_id": "294c51a0-ecba-4b4a-8ba6-14eb46f36551", "name": "Kenneth Ross", "description": "Direction partner free. Friend history leader no drive main later.\nDnour put o",
  "_id": "800eed73-54df-4cc9-841a-eb7f54945926", "name": "Michelle Davis", "description": "Candidate something leg leader week culture check. Step law store reveal wr",
  "_id": "5bc55653-8a54-4781-8a6b-a8a57d24348b", "name": "Phillip Robinson", "description": "Ago coach continue space product fine. Wear little story responsibility b",
  "_id": "850d5a7c-5a2b-4701-8380-69f1e957e4da", "name": "Beth Wilson", "description": "Follow shake pay as. Conference here pull yes able against field.\nDirection a",
  "_id": "df64dc7c-6993-481a-b729-f7900334d9c1", "name": "Jay Page", "description": "Egal analysis structure despite. Their floor war face call major true.", "category",
  "_id": "c388a03a-e094-4b51-9b7d-b7f12c0219f2", "name": "Melissa Hernandez", "description": "Sometimes candidate security number trade help. Retail knowledge former",
  "_id": "d96ce1fd-6168-4a8b-a51b-1c33a011ac5", "name": "Charles White", "description": "Technology today stand five show hair force. Similar require sometimes two l",
  "_id": "f14f10e8-d3f7-4d05-be54-ca08a206347f", "name": "Jose Parrish", "description": "Show assume scientist box argue issue none. Sign myself paper world possible",
  "_id": "5433080b-9824-4c0e-b196-7eb6423bf8b8", "name": "Amanda Smith", "description": "Republican Mr free money often fund thank. Listen tonight health might serio",
  "_id": "59df985a-791a-4bb6-9505-c86a67e888ac", "name": "Crystal Pearson", "description": "Citizen so energy participant serious.\nDnll out person. Understand safe wo",
  "_id": "692fa278-b04c-45e8-ac1f-f7c2e34ef339", "name": "Paul Moore", "description": "Show decade pressure general others fall. Action system sometimes away.\nDnrow e",
  "_id": "e8a7e1c2-df20-4b31-95a2-d99e272e9dfc", "name": "Linda Diaz", "description": "Blood nest ego itself no store look include. Build left wait.\nLay bag group st",
  "_id": "48d294c2-8a22-480a-ba05-11a3c8bb0da", "name": "Michael Williams", "description": "Left past explain. Then past become you message righte kind.\nDnmond you",
  "_id": "57d6d789-fd63-4b1c-8198-63c2c03d3d65", "name": "Thomas Dennis", "description": "Traditional these protect. Grow prepare others stand mission. Dream ciala ow",
  "_id": "ab13747e-b24c-44aa-83bf-48f6bc4c0b4d", "name": "Christopher Harris", "description": "Enough include such per not then system goal. Participant book natural",
  "_id": "4d6888cb-fd05-4022-ae78-9eddfcd85935", "name": "Leah Faulkner", "description": "Environmental per family group people. Film study nearly health.\nDnrow durin",
  "_id": "65bcb491-ba93-4d96-9318-6aa4f892323c", "name": "Jill Thompson", "description": "According behind specific tough protect. Piece size wear add.\nDnforget cause",
  "_id": "d96eabbf-4d8a-49a1-84dc-df9efaf69f69", "name": "Jill Malone", "description": "Dog require shake history save buy interview.\nDnbecome eye everybody join prote",
  "_id": "d49eaf1f-2943-4dc8-badb-790d75b61c32", "name": "Megan Reeves", "description": "Officer tend turn while. South poor crime receive. Eye bring site.\nDneconomy f",
  "_id": "d5724df3-a108-49c8-90d2-d9761a6d5d86", "name": "Regina May", "description": "Skill many should throw specific rather memory management. Respond life right a",
  "_id": "d65c33a5-230f-4a0a-b39f-4d0af6313335", "name": "Jonathan Powell", "description": "Charge cause through speech everything. Wind discussion this.\nDnthough sit",
  "_id": "c11a155a-2f35-44dc-8149-a054b0bf0bdc", "name": "Tiffany Andrews", "description": "Happen book professional check. Develop use mother score my seek.\nDnline fu",
  "_id": "df8c3550-b971-40ff-badd-841aa67c60b", "name": "Shari Burnett", "description": "Someone painting poor.\nDnreport teach could woman wait picture gun. History f",
  "measure the time with no index: 0.2270s"
}
Process finished with exit code 0
```

Figure 16: Test 4

```
Run test5
{
  "_id": "e8b7142b-e177-49a3-b727-2a9f808a6a62", "name": "Taylor Clark", "description": "Bar economic stand clearly. Kind unit from son take administration. Owner lay",
  "_id": "ffefcd24-7b21-4764-90d0-84b385747c49", "name": "Sara Price", "description": "Ok win view two pull hope answer reveal. Movement buy then house address gener",
  "_id": "ea0b10af-0942-43f8-9a35-6411c772b6dd", "name": "Martha Thomas", "description": "Box suffer film company. House national ahead wind seven deal fall keep. Use",
  "_id": "866455c1-8151-4a84-a00a-9d4d2b6c09e4", "name": "Dawn Lester", "description": "Center energy still development. Response administration agent management buy",
  "_id": "c61625f1-84f2-432a-aa98-f1c0dcac9c6b", "name": "Joseph Coleman", "description": "Ball fill summer stay. Sister son good contain everyone property sit admini",
  "_id": "1aedca50-4caa-4ccf-bba4-262d3a61277f", "name": "Maria Irwin MD", "description": "Place high rest manage. Foot you already learn owner take administration.\n",
  "_id": "aac4bf03-1525-4547-8499-1336df9eaff8", "name": "Nancy Perez", "description": "Factor tax seat per drop trial war. Administration month loss region none sol",
  "_id": "ba37f0ba-430f-4a0a-b39f-4d0af6313335", "name": "Michael Brennan", "description": "Reality card fill believe administration. Message fire per student free as",
  "_id": "f551b876-9aeb-4d25-8887-b6a6d98a747f", "name": "Megan Jones", "description": "What treatment chance live region cover lat. Policy say eastern bad news.\nAda",
  "_id": "698a7a58-5d59-4737-835d-e096d38b5a0f", "name": "David Suzman", "description": "Success wind match south. A long public your people.\nLefst tree wrong. None f",
  "_id": "d7959437-8a3e-4a95-bc86-df1c12901498", "name": "Patricia Sanchez", "description": "Employee Republican bit care ability lay arrive.\nDnserve social medical cc",
  "_id": "e683a488-ea1d-4026-aa9c-bd2355427a0c", "name": "David Yu", "description": "Early result since treat star. Draw feel check loss decade office matter.\nDnll tr",
  "_id": "8e47b7e9-6a1f-4c10-ac12-6920f9c9153", "name": "Victoria Davis", "description": "Go become moment allow. Person yet mention individual. Nature say bad somet",
  "_id": "85cd3c7c-9342-48d9-9747-168366110ba3", "name": "Donna Moore", "description": "Church compare final treat. Join response later manage can true know prepare.\n",
  "_id": "ae6a809c-ae7b-4787-b5d2-80fbd314c411", "name": "Paul Leblanc", "description": "He staff hit east baby. Choose forward owner contain or administration low. A",
  "_id": "76784a55-0537-4a09-992f-83c8a4ca2095", "name": "Jonathan Frazer", "description": "Put lay program southern produce. Front seek hear yes idea pull administ",
  "_id": "8a4f5969-3c1b-4b27-8186-fff369300821", "name": "Michael Forbes", "description": "Site offer none hard change condition country TV. Single join break gas up",
  "_id": "5acbbd21-a380-4633-acc1-a652b3797011", "name": "Christopher Lynn", "description": "Hotel low goal doctor. Feel ahead inside budget group. Candidate laugh cl",
  "_id": "c7907acc-ea1b-4d44-9994-af65b2c89e2c", "name": "Gabrielle Thompson", "description": "Part determines concern player road cup. Speech hair perform heavy hit.",
  "_id": "08d8d6df-646b-470f-8db1-505e4b669a1d", "name": "David Reyes", "description": "Stand stand outside stuff moment. Prevent wind case say boy political. Came th",
  "_id": "23d2a8bf-1cae-42f7-8b84-a168d3656b92", "name": "Jessica Eaton", "description": "President now discover war film kitchen radio see. West build total kid. Way",
  "_id": "2d2adab8-9e1f-4c9f-b16d-7f68d276192b", "name": "Keith Garcia", "description": "Alone bag a coach minute. Argue drug theory cover move may. Example want kid",
  "_id": "c74e3be9-f928-4aaa-ae67-fe4a9a31831a", "name": "Olivia Maynard", "description": "Black specific may rock. Benefit number want home end. Two rock wait seen p",
  "current index",
  "SON([('v', 2), ('key', SON([('id', 1)])), ('name', '_id')])",
  "SON([('v', 2), ('key', SON([('location', '2dsphere')]), ('name', 'Location.2dsphere'), ('2dsphereIndexVersion', 3)])",
  "SON([('v', 2), ('key', SON([('fts', 'text'), ('_ftsx', 1)])), ('name', 'description_text'), ('weights', SON([('description', 1)])), ('default_language', 'english')]",
  "Process finished with exit code 0
```

Figure 17: Test 5