CSCI 5408

DATA MANAGEMENT AND WAREHOUSING

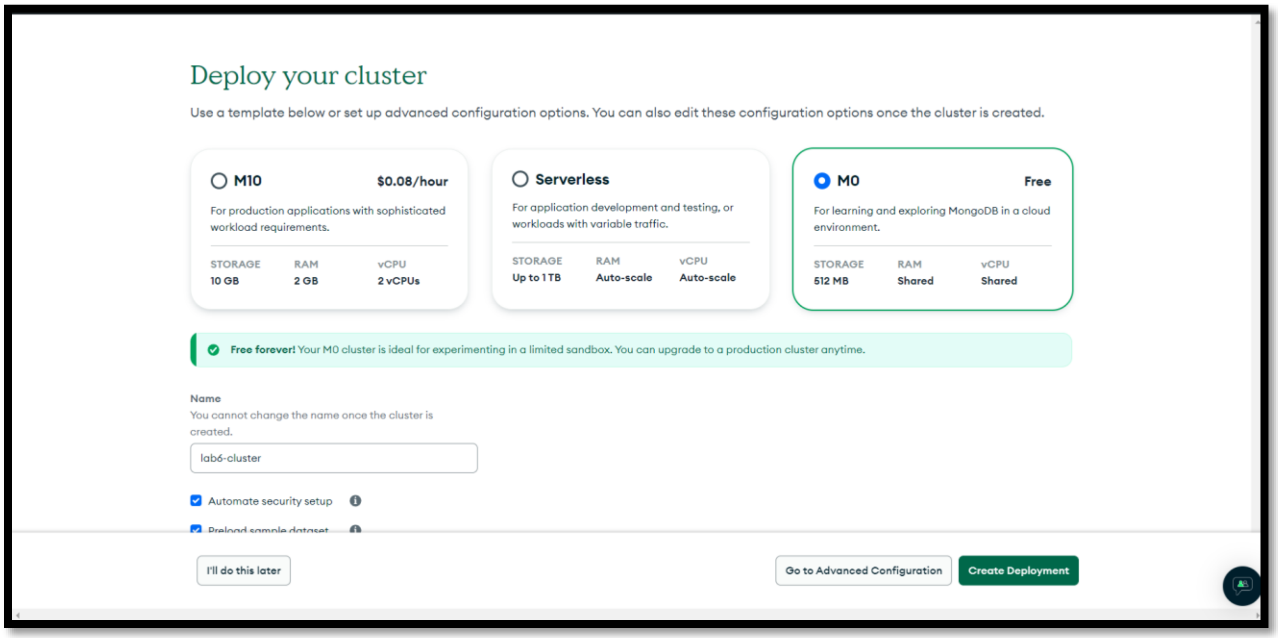
LAB - 6

Banner ID: B00984406
GitLab Assignment Link:
https://git.cs.dal.ca/jems/csci5408_s24_b00984406_jems_patel.git

Table of Contents

Task1: MongoDB……………………………………………………………………………………3
References…………………………………………………………………………………………12

Task 1: MongoDB:



Deploy your cluster

Use a template below or set up advanced configuration options. You can also edit these configuration options once the cluster is created.

[ ] M10 — $0.08/hour
For production applications with sophisticated workload requirements.
STORAGE 10 GB · RAM 2 GB · vCPU 2 vCPUs

[ ] Serverless
For application development and testing, or workloads with variable traffic.
STORAGE Up to 1 TB · RAM Auto-scale · vCPU Auto-scale

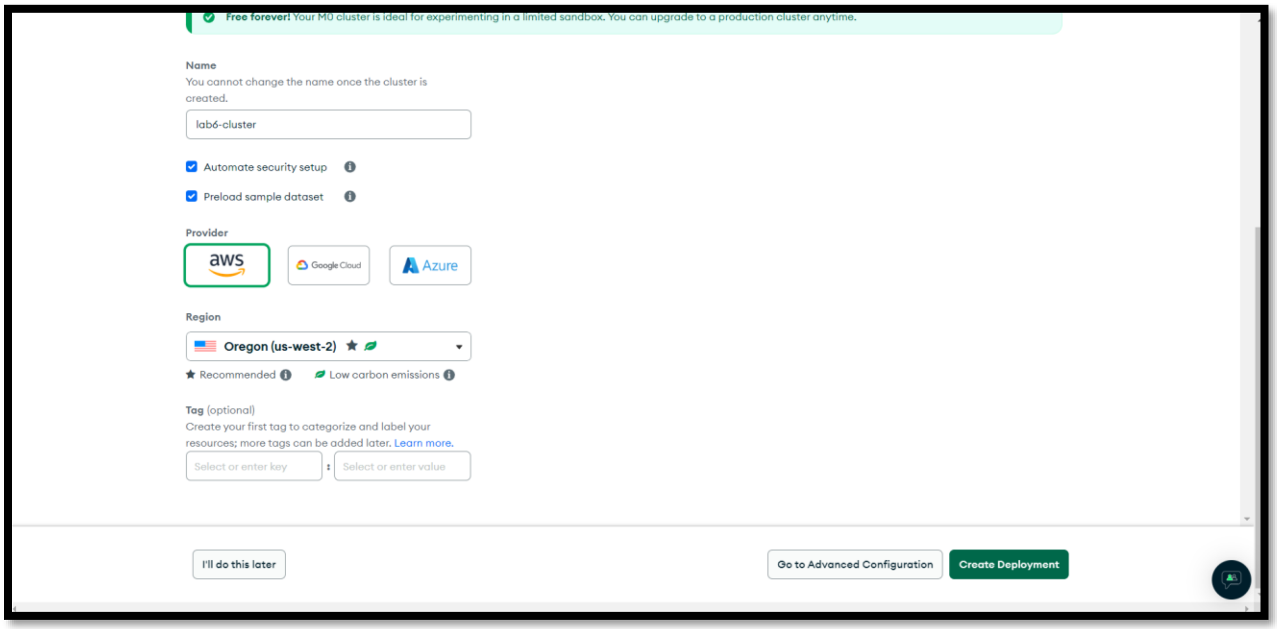
[✓ M0] — Free
For learning and exploring MongoDB in a cloud environment.
STORAGE 512 MB · RAM Shared · vCPU Shared

Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Name
You cannot change the name once the cluster is created.
lab6-cluster

☑ Automate security setup
☑ Preload sample dataset

Figure 1.1: Create a Cluster of name "lab6-cluster"



Free forever! Your M0 cluster is ideal for experimenting in a limited sandbox. You can upgrade to a production cluster anytime.

Name
You cannot change the name once the cluster is created.
lab6-cluster

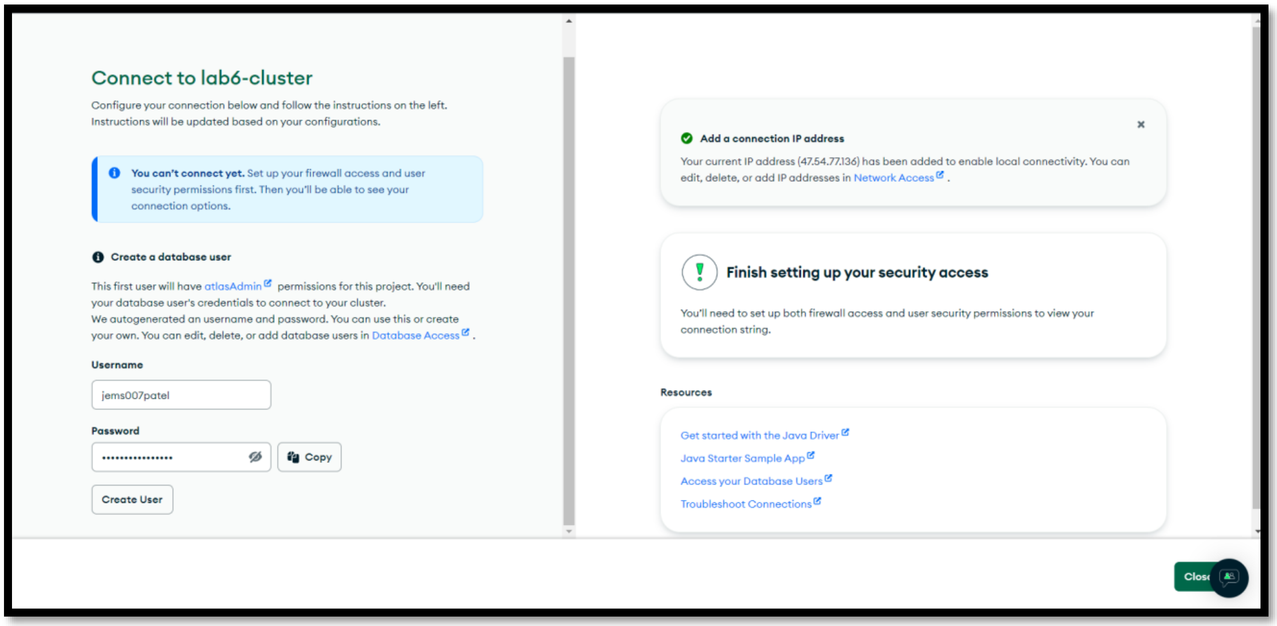
☑ Automate security setup
☑ Preload sample dataset

Provider
[✓ aws] Google Cloud Azure

Region
Oregon (us-west-2) ★
★ Recommended · Low carbon emissions

Tag (optional)
Create your first tag to categorize and label your resources; more tags can be added later. Learn more.
Select or enter key : Select or enter value

Figure 1.2: Choose cloud provider and region



Connect to lab6-cluster

Configure your connection below and follow the instructions on the left. Instructions will be updated based on your configurations.

You can't connect yet. Set up your firewall access and user security permissions first. Then you'll be able to see your connection options.

Create a database user

This first user will have atlasAdmin permissions for this project. You'll need your database user's credentials to connect to your cluster.
We autogenerated an username and password. You can use this or create your own. You can edit, delete, or add database users in Database Access .

Username: jems007patel

Password: ••••••••••••••••

Add a connection IP address

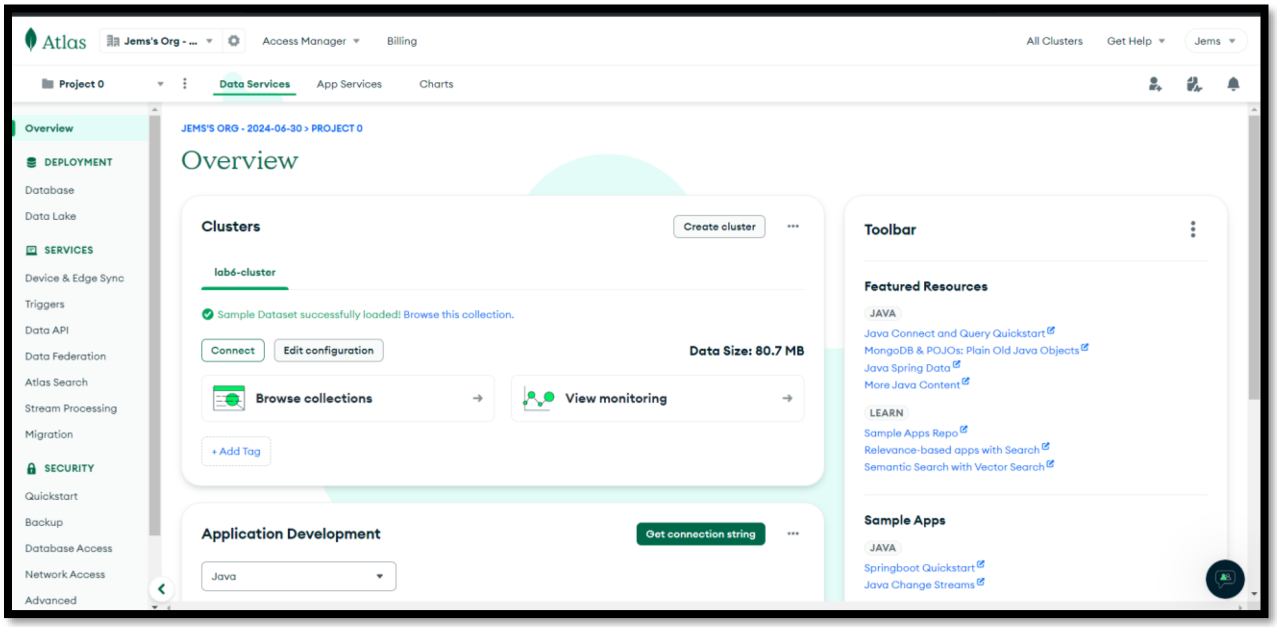
Your current IP address (47.54.77.136) has been added to enable local connectivity. You can edit, delete, or add IP addresses in Network Access .

Finish setting up your security access

You'll need to set up both firewall access and user security permissions to view your connection string.

Get started with the Java Driver
Java Starter Sample App
Access your Database Users
Troubleshoot Connections

Figure 1.3: Set up a new user



JEMS'S ORG - 2024-06-30 > PROJECT 0

Overview

Clusters

lab6-cluster

Sample Dataset successfully loaded! Browse this collection.

Data Size: 80.7 MB

Application Development

Toolbar

Featured Resources

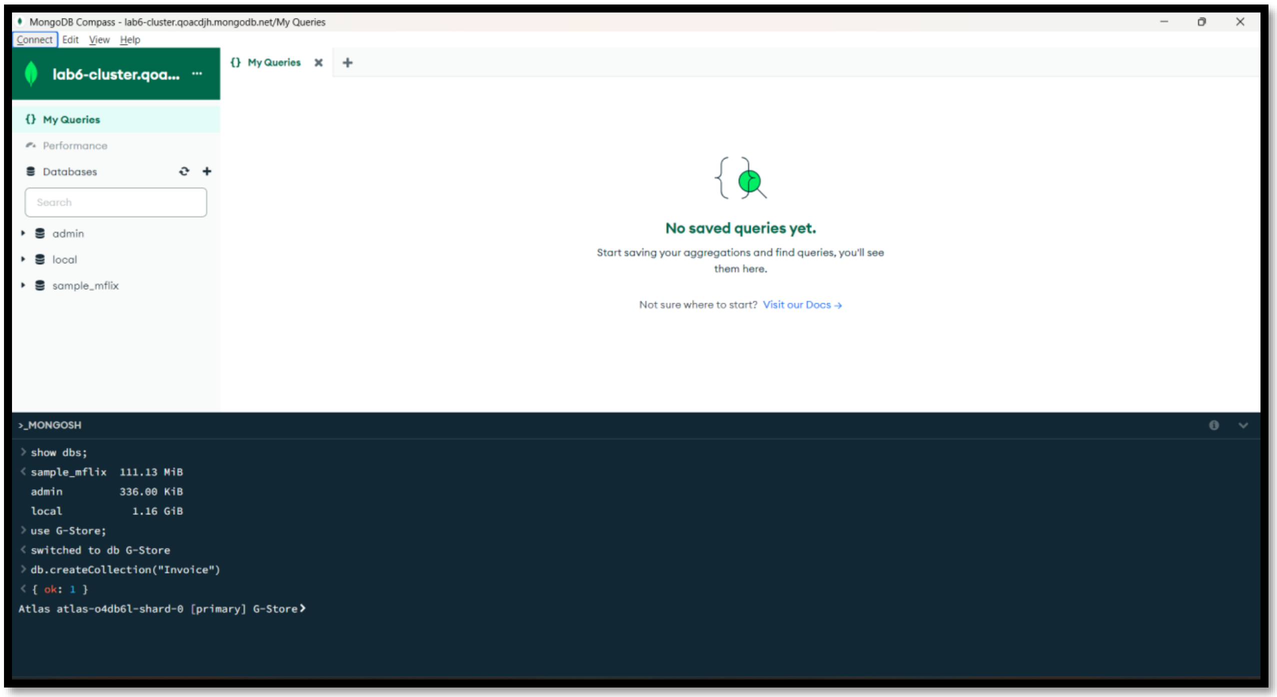
JAVA
Java Connect and Query Quickstart
MongoDB & POJOs: Plain Old Java Objects
Java Spring Data
More Java Content

LEARN
Sample Apps Repo
Relevance-based apps with Search
Semantic Search with Vector Search

Sample Apps

JAVA
Springboot Quickstart
Java Change Streams

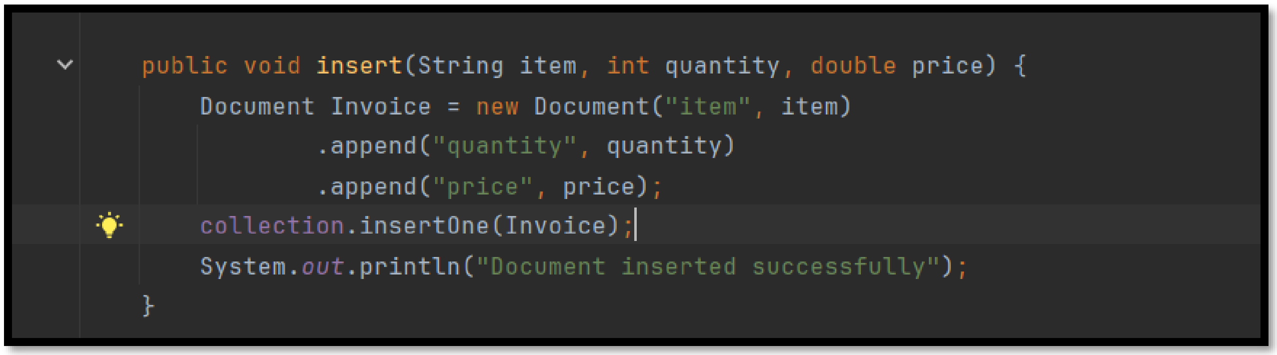
Figure 1.4: Cluster creation is successfully done.


MongoDB Compass - lab6-cluster.qoacdjh.mongodb.net/My Queries
Connect
Edit
View
Help
lab6-cluster.qoa...
My Queries
Performance
Databases
Search
admin
local
sample_mflix
No saved queries yet.
Start saving your aggregations and find queries, you'll see them here.
Not sure where to start? Visit our Docs →
>_MONGOSH

```
> show dbs;
< sample_mflix  111.13 MiB
  admin         336.00 KiB
  local           1.16 GiB
> use G-Store;
< switched to db G-Store
> db.createCollection("Invoice")
< { ok: 1 }
Atlas atlas-o4db6l-shard-0 [primary] G-Store>
```

Figure 1.5: Create database and collection in MongoDB Compass

INSERT:



```java
public void insert(String item, int quantity, double price) {
    Document Invoice = new Document("item", item)
            .append("quantity", quantity)
            .append("price", price);
    collection.insertOne(Invoice);
    System.out.println("Document inserted successfully");
}
```

Figure 1.6: Code of the insert query

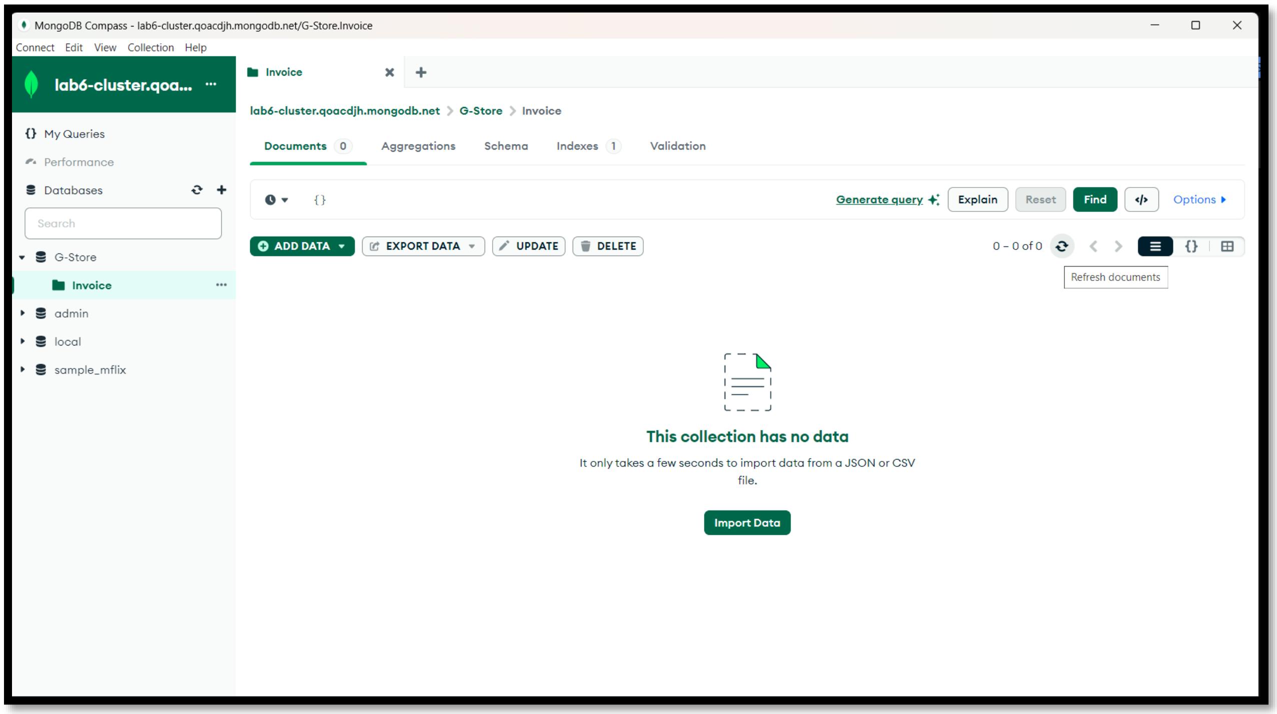
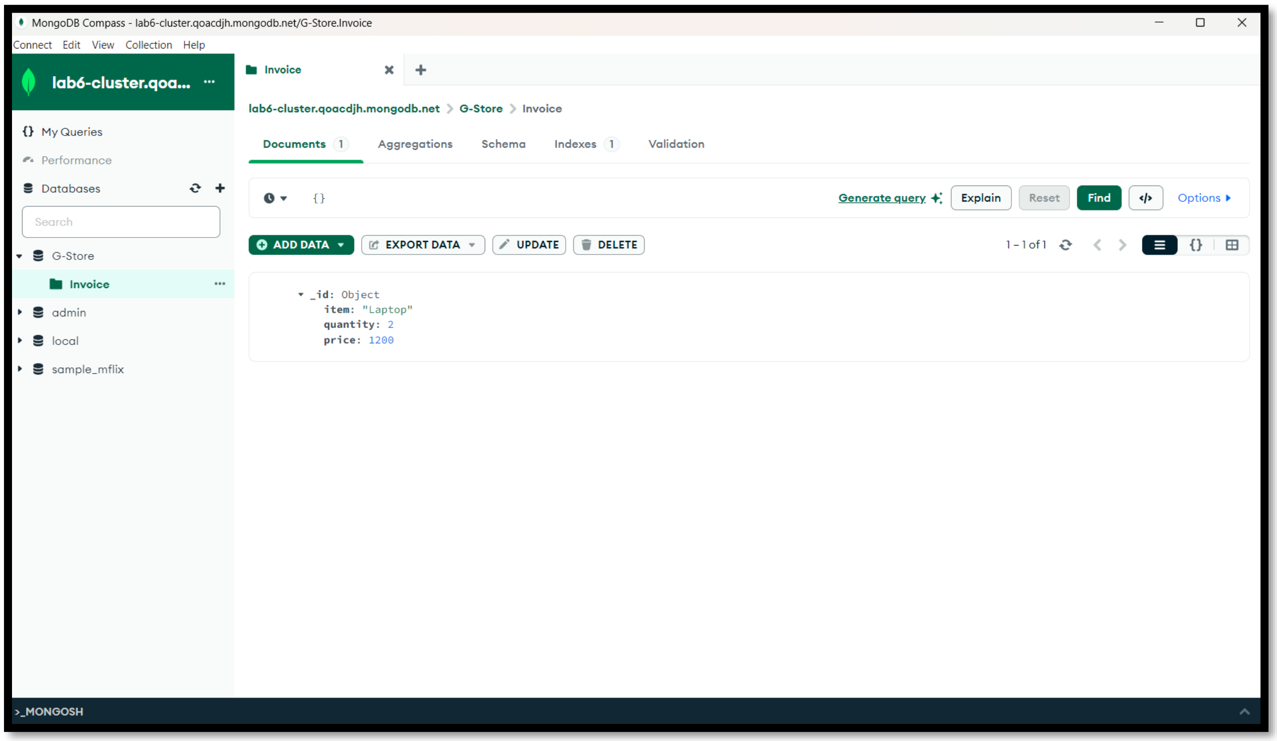


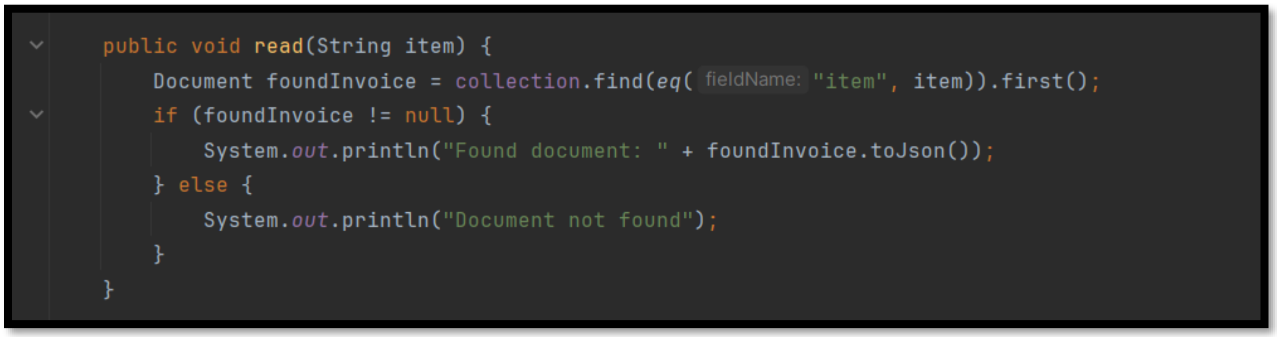
Figure 1.7: Database before insert query performed



```
▾ _id: Object
  item: "Laptop"
  quantity: 2
  price: 1200
```

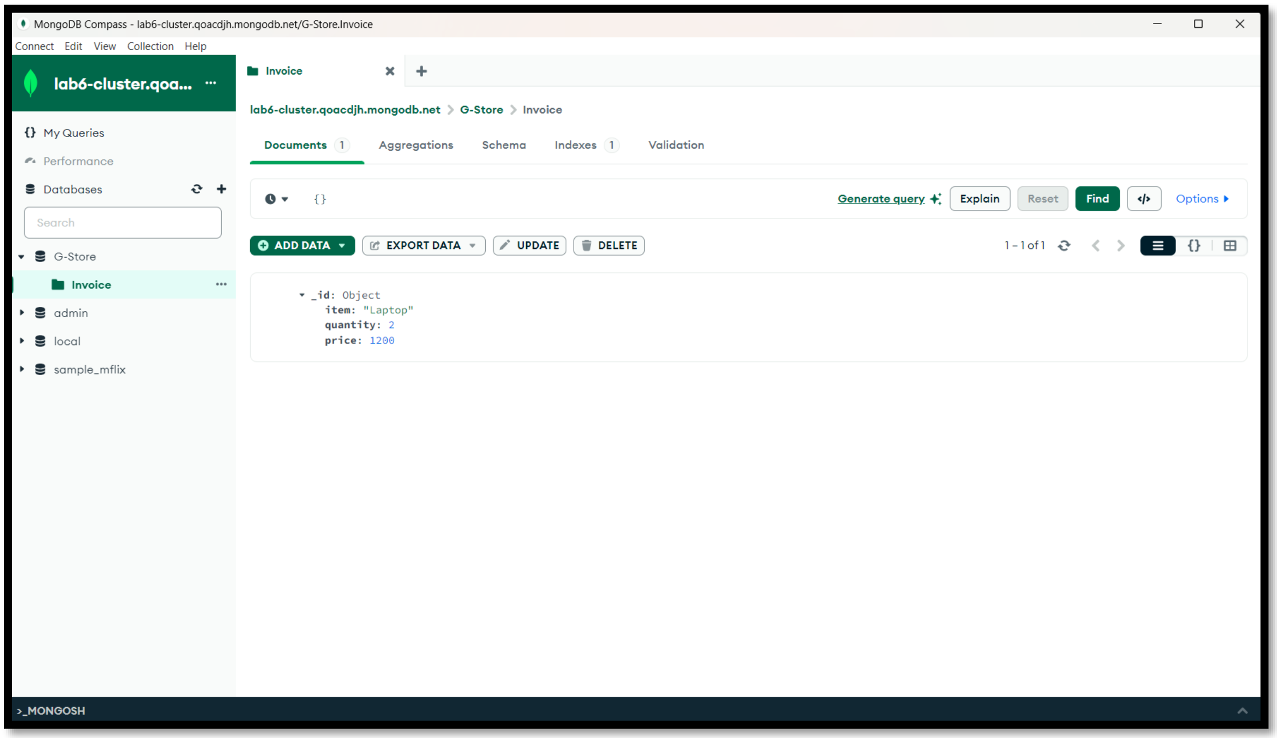
Figure 1.8: Database after Insert query performed

READ:



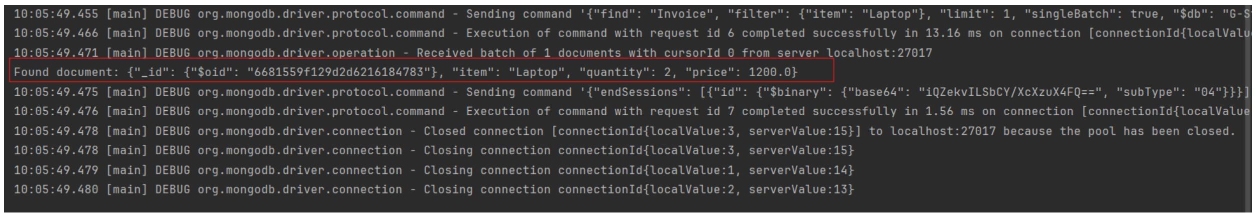
```
public void read(String item) {
    Document foundInvoice = collection.find(eq( fieldName: "item", item)).first();
    if (foundInvoice != null) {
        System.out.println("Found document: " + foundInvoice.toJson());
    } else {
        System.out.println("Document not found");
    }
}
```

Figure 1.9: Read database code



```
_id: Object
item: "Laptop"
quantity: 2
price: 1200
```

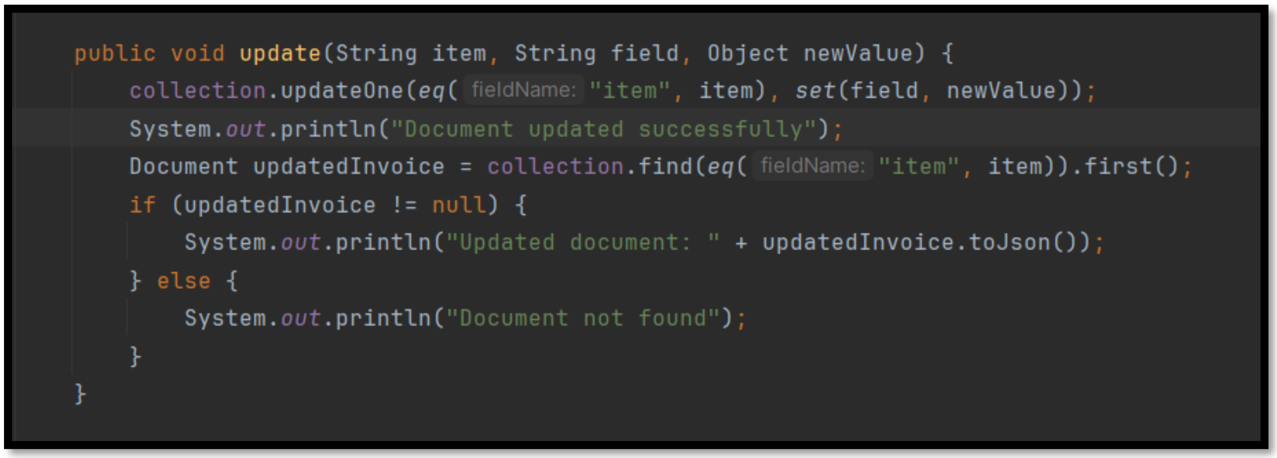
Figure 1.10: Database before the query performed



```
10:05:49.455 [main] DEBUG org.mongodb.driver.protocol.command - Sending command '{"find": "Invoice", "filter": {"item": "Laptop"}, "limit": 1, "singleBatch": true, "$db": "G-S
10:05:49.466 [main] DEBUG org.mongodb.driver.protocol.command - Execution of command with request id 6 completed successfully in 13.16 ms on connection [connectionId{localValu
10:05:49.471 [main] DEBUG org.mongodb.driver.operation - Received batch of 1 documents with cursorId 0 from server localhost:27017
Found document: {"_id": {"$oid": "6681559f129d2d6216184783"}, "item": "Laptop", "quantity": 2, "price": 1200.0}
10:05:49.475 [main] DEBUG org.mongodb.driver.protocol.command - Sending command '{"endSessions": [{"id": {"$binary": {"base64": "iQZekvILSbCY/XcXzuX4FQ==", "subType": "04"}}}]
10:05:49.476 [main] DEBUG org.mongodb.driver.protocol.command - Execution of command with request id 7 completed successfully in 1.56 ms on connection [connectionId{localValue
10:05:49.478 [main] DEBUG org.mongodb.driver.connection - Closed connection [connectionId{localValue:3, serverValue:15}] to localhost:27017 because the pool has been closed.
10:05:49.478 [main] DEBUG org.mongodb.driver.connection - Closing connection connectionId{localValue:3, serverValue:15}
10:05:49.479 [main] DEBUG org.mongodb.driver.connection - Closing connection connectionId{localValue:1, serverValue:14}
10:05:49.480 [main] DEBUG org.mongodb.driver.connection - Closing connection connectionId{localValue:2, serverValue:13}
```

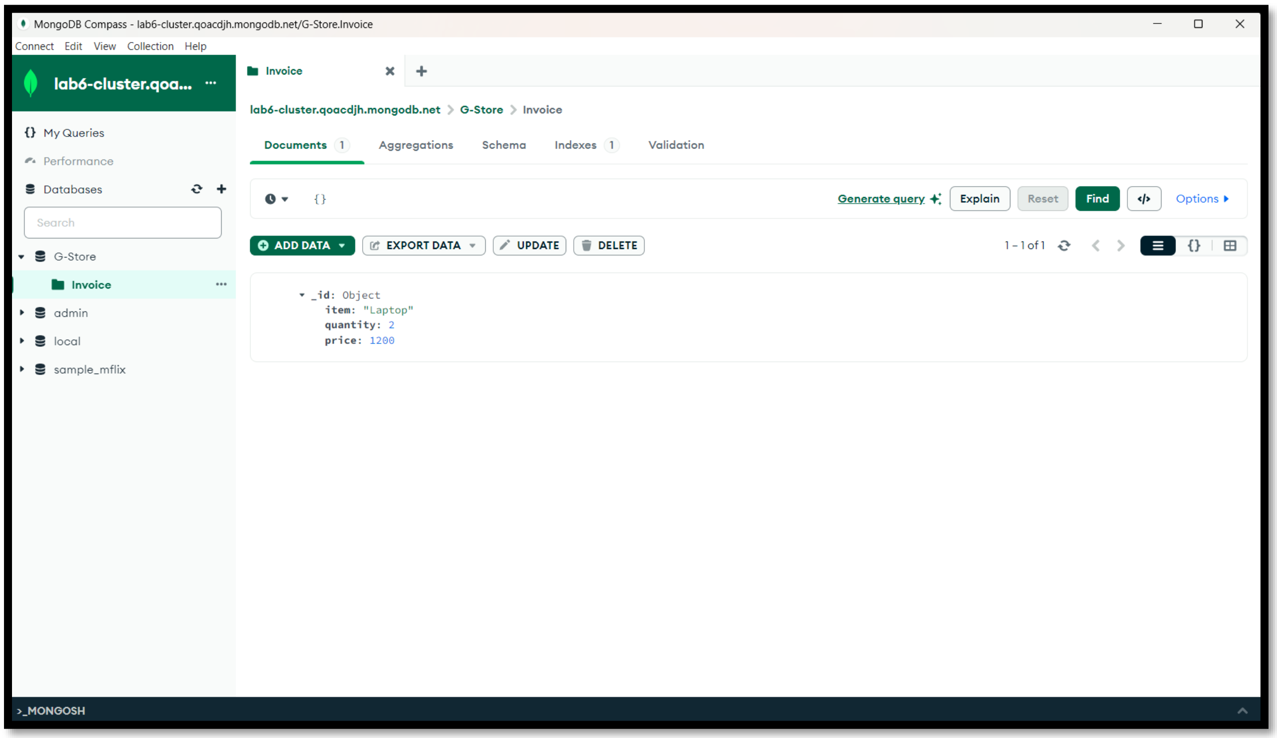
Figure 1.11: Run READ query

UPDATE:



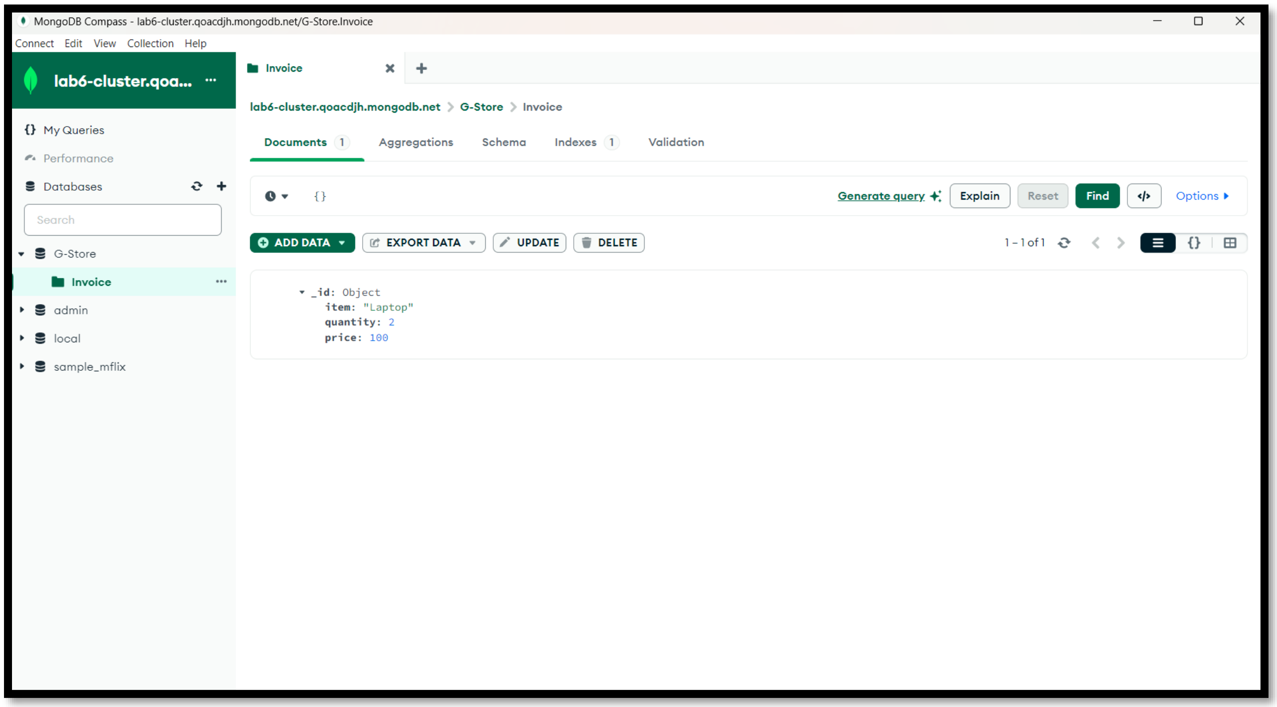
```java
public void update(String item, String field, Object newValue) {
    collection.updateOne(eq( fieldName: "item", item), set(field, newValue));
    System.out.println("Document updated successfully");
    Document updatedInvoice = collection.find(eq( fieldName: "item", item)).first();
    if (updatedInvoice != null) {
        System.out.println("Updated document: " + updatedInvoice.toJson());
    } else {
        System.out.println("Document not found");
    }
}
```

Figure 1.12: Update database code



```
_id: Object
item: "Laptop"
quantity: 2
price: 1200
```

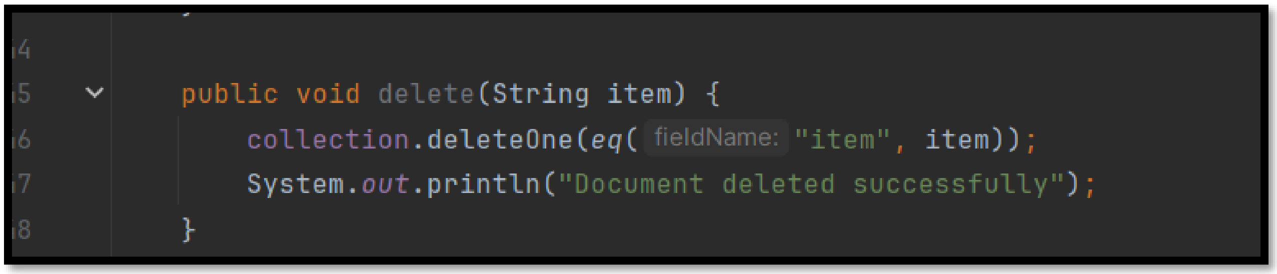
Figure 1.13: Database before the update query performed



```
_id: Object
item: "Laptop"
quantity: 2
price: 100
```

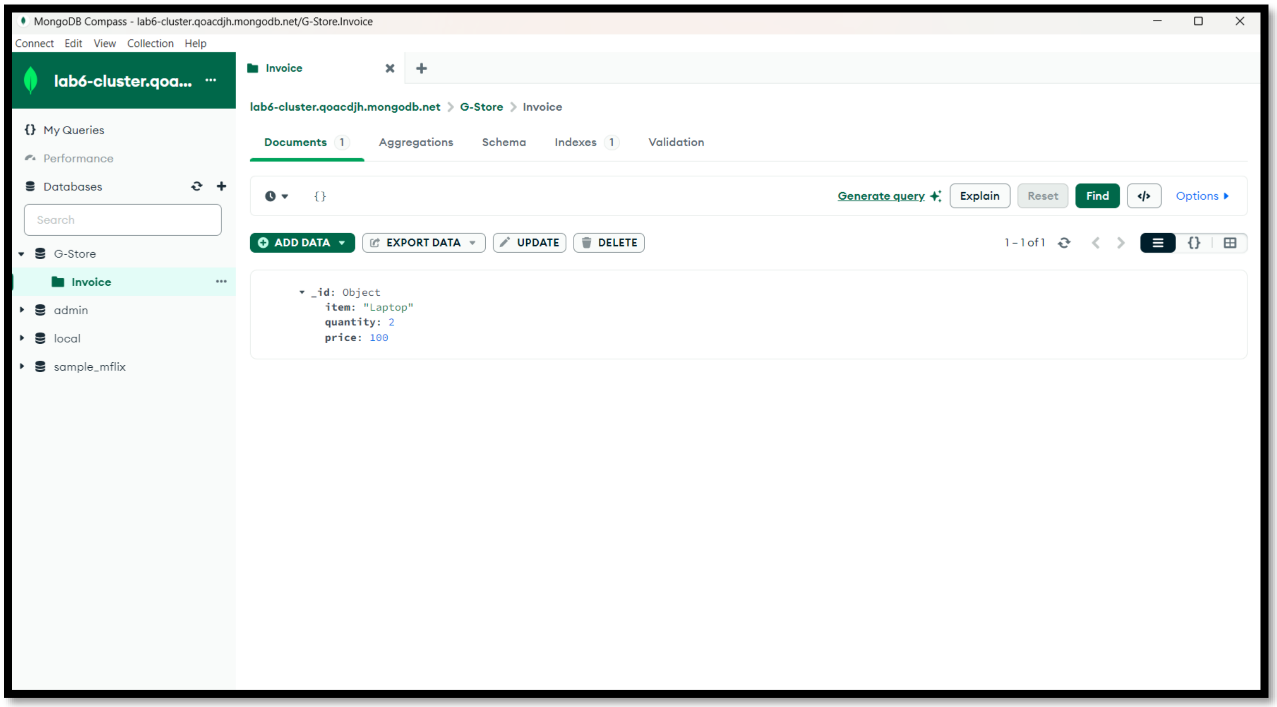
Figure 1.14: Database after the update query performed

DELETE:



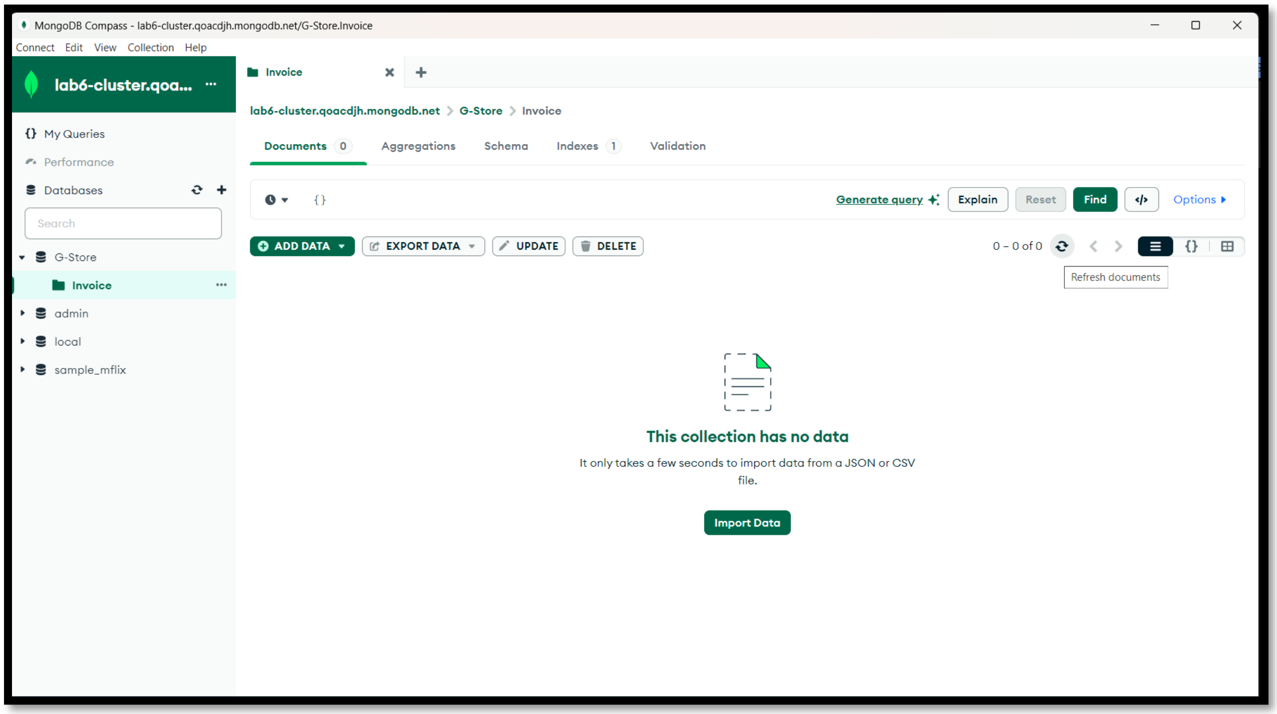
```java
public void delete(String item) {
    collection.deleteOne(eq( fieldName: "item", item));
    System.out.println("Document deleted successfully");
}
```

Figure 1.15: Delete query code



```
▾ _id: Object
  item: "Laptop"
  quantity: 2
  price: 100
```

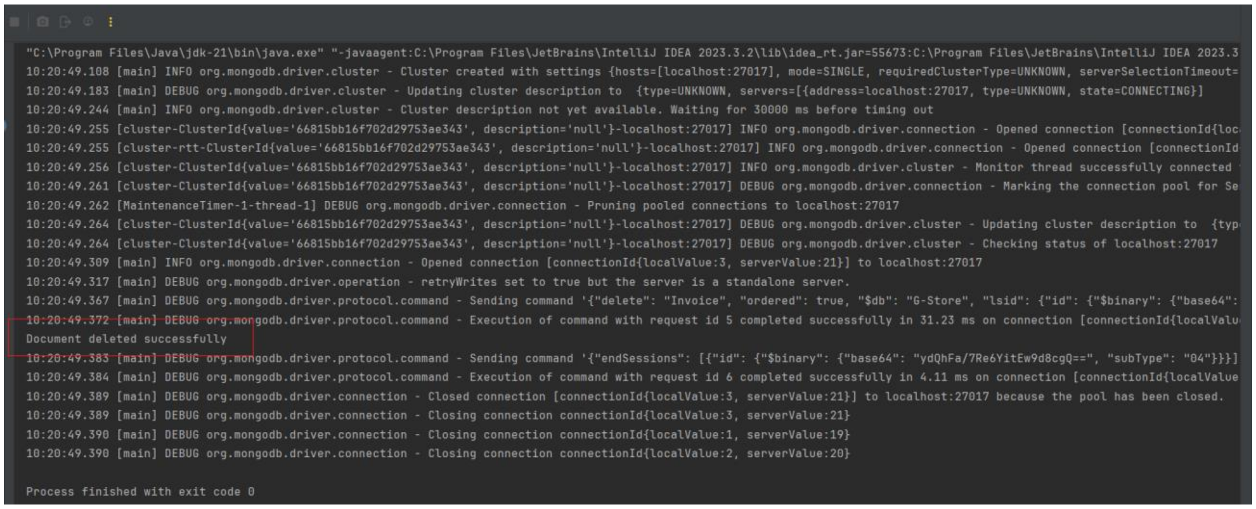
Figure 1.16: Database before the delete query performed



This collection has no data

It only takes a few seconds to import data from a JSON or CSV file.

Figure 1.17: Database after the delete query performed



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3.2\lib\idea_rt.jar=55673:C:\Program Files\JetBrains\IntelliJ IDEA 2023.3
10:20:49.108 [main] INFO org.mongodb.driver.cluster - Cluster created with settings {hosts=[localhost:27017], mode=SINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout=
10:20:49.183 [main] DEBUG org.mongodb.driver.cluster - Updating cluster description to  {type=UNKNOWN, servers=[{address=localhost:27017, type=UNKNOWN, state=CONNECTING}]
10:20:49.244 [main] INFO org.mongodb.driver.cluster - Cluster description not yet available. Waiting for 30000 ms before timing out
10:20:49.255 [cluster-ClusterId{value='66815bb16f702d29753ae343', description='null'}-localhost:27017] INFO org.mongodb.driver.connection - Opened connection [connectionId{loc
10:20:49.255 [cluster-rtt-ClusterId{value='66815bb16f702d29753ae343', description='null'}-localhost:27017] INFO org.mongodb.driver.connection - Opened connection [connectionId
10:20:49.256 [cluster-ClusterId{value='66815bb16f702d29753ae343', description='null'}-localhost:27017] INFO org.mongodb.driver.cluster - Monitor thread successfully connected
10:20:49.261 [cluster-ClusterId{value='66815bb16f702d29753ae343', description='null'}-localhost:27017] DEBUG org.mongodb.driver.connection - Marking the connection pool for Se
10:20:49.262 [MaintenanceTimer-1-thread-1] DEBUG org.mongodb.driver.connection - Pruning pooled connections to localhost:27017
10:20:49.264 [cluster-ClusterId{value='66815bb16f702d29753ae343', description='null'}-localhost:27017] DEBUG org.mongodb.driver.cluster - Updating cluster description to  {typ
10:20:49.264 [cluster-ClusterId{value='66815bb16f702d29753ae343', description='null'}-localhost:27017] DEBUG org.mongodb.driver.cluster - Checking status of localhost:27017
10:20:49.309 [main] INFO org.mongodb.driver.connection - Opened connection [connectionId{localValue:3, serverValue:21}] to localhost:27017
10:20:49.317 [main] DEBUG org.mongodb.driver.operation - retryWrites set to true but the server is a standalone server.
10:20:49.367 [main] DEBUG org.mongodb.driver.protocol.command - Sending command '{"delete": "Invoice", "ordered": true, "$db": "G-Store", "lsid": {"id": {"$binary": {"base64":
10:20:49.372 [main] DEBUG org.mongodb.driver.protocol.command - Execution of command with request id 5 completed successfully in 31.23 ms on connection [connectionId{localValu
Document deleted successfully
10:20:49.383 [main] DEBUG org.mongodb.driver.protocol.command - Sending command '{"endSessions": [{"id": {"$binary": {"base64": "ydQhFa/7Re6YitEw9d8cgQ==", "subType": "04"}}}]
10:20:49.384 [main] DEBUG org.mongodb.driver.protocol.command - Execution of command with request id 6 completed successfully in 4.11 ms on connection [connectionId{localValue
10:20:49.389 [main] DEBUG org.mongodb.driver.connection - Closed connection [connectionId{localValue:3, serverValue:21}] to localhost:27017 because the pool has been closed.
10:20:49.389 [main] DEBUG org.mongodb.driver.connection - Closing connection connectionId{localValue:3, serverValue:21}
10:20:49.390 [main] DEBUG org.mongodb.driver.connection - Closing connection connectionId{localValue:1, serverValue:19}
10:20:49.390 [main] DEBUG org.mongodb.driver.connection - Closing connection connectionId{localValue:2, serverValue:20}

Process finished with exit code 0
```

Figure 1.18: Run the delete query

References

[1] "How to connect MongoDB with Java program?," Stack Overflow, Available: https://stackoverflow.com/questions/52621363/how-to-connect-mongodb-with-java-program. [Accessed: Jun. 30, 2024].

[2] "MongoDB Fundamentals - CRUD," MongoDB, Available: https://www.mongodb.com/resources/products/fundamentals/crud. [Accessed: Jun. 30, 2024].

[3] "MongoDB Java Driver Documentation," MongoDB, Available: https://www.mongodb.com/docs/drivers/java/sync/current/. [Accessed: Jun. 30, 2024].