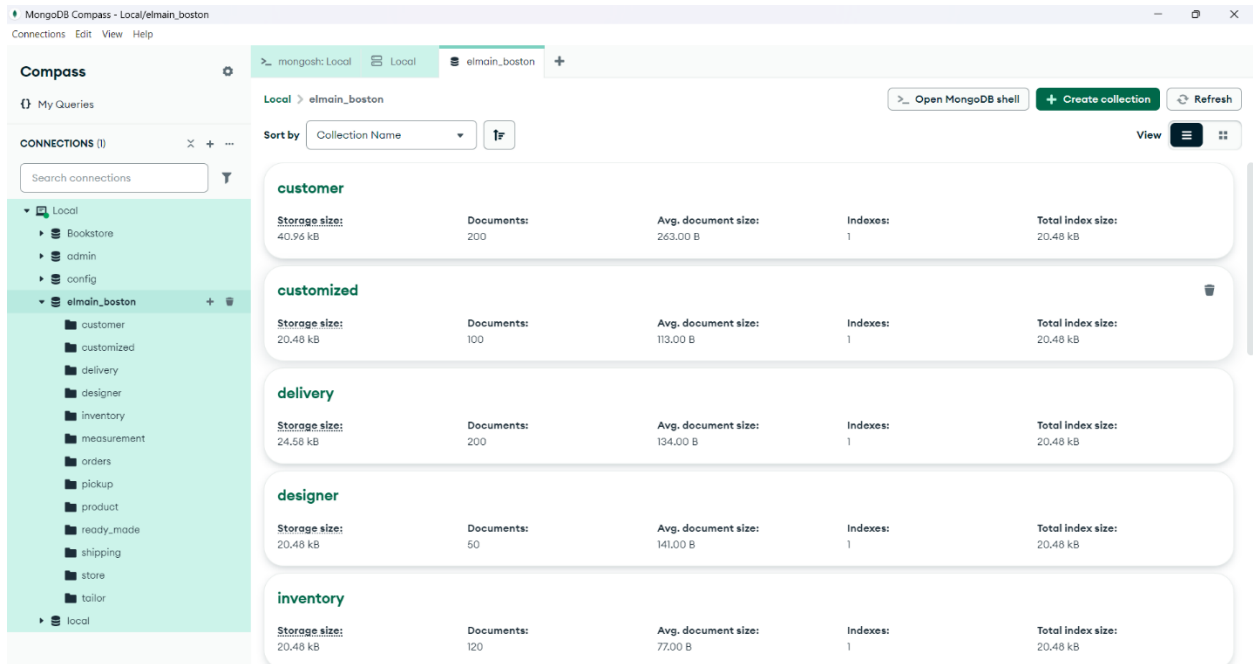


## Python code to implement elmain\_boston database in MongoDB

```
2      """
3      Created on Wed Nov 27 16:28:16 2024
4
5      @author: ragul
6      """
7
8      import mysql.connector
9      import pymongo
10     from decimal import Decimal
11     from datetime import datetime, date, timedelta
12
13     # Define the function to convert datetime.date, decimal.Decimal, and timedelta objects
14     def convert_date_objects(obj):
15         if isinstance(obj, date): # If the object is a datetime.date instance
16             return datetime.combine(obj, datetime.min.time()) # Convert to datetime.datetime
17         elif isinstance(obj, timedelta): # If the object is a datetime.timedelta instance
18             return obj.total_seconds() # Convert to total number of seconds
19         elif isinstance(obj, Decimal): # If the object is a decimal.Decimal
20             return float(obj) # Convert to float
21         elif isinstance(obj, dict): # If the object is a dictionary, recurse through it
22             return {k: convert_date_objects(v) for k, v in obj.items()}
23         elif isinstance(obj, list): # If the object is a list, recurse through it
24             return [convert_date_objects(item) for item in obj]
25         else:
26             return obj # Return other types unchanged
27
28     # Connect to MySQL database
29     mysql_conn = mysql.connector.connect(
30         host="127.0.0.1", # MySQL server
31         user="root", # MySQL username
32         password="Ie6700", # MySQL password
33         database="elmain_boston"
34     )
35     cursor = mysql_conn.cursor()
36
37     # Connect to MongoDB
38     mongo_client = pymongo.MongoClient("mongodb://localhost:27017/")
39     mongo_db = mongo_client["elmain_boston"]
40
41     # Get tables from MySQL
42     cursor.execute("SHOW TABLES")
43     tables = cursor.fetchall()
44
45     for table in tables:
46         table_name = table[0]
47         cursor.execute(f"SELECT * FROM {table_name}")
48         rows = cursor.fetchall()
49         columns = [description[0] for description in cursor.description]
50
51         # Convert rows to dictionaries
52         data = []
53         for row in rows:
54             row_data = dict(zip(columns, row))
55
56             # Step 1: Convert any datetime.date, datetime.timedelta, or decimal.Decimal to MongoDB-compatible types
57             row_data = convert_date_objects(row_data)
58
59             data.append(row_data)
60
61         # Step 2: Insert into MongoDB collection
62         mongo_collection = mongo_db[table_name]
63         mongo_collection.insert_many(data)
64
65     # Close the MySQL connection
66     mysql_conn.close()
```

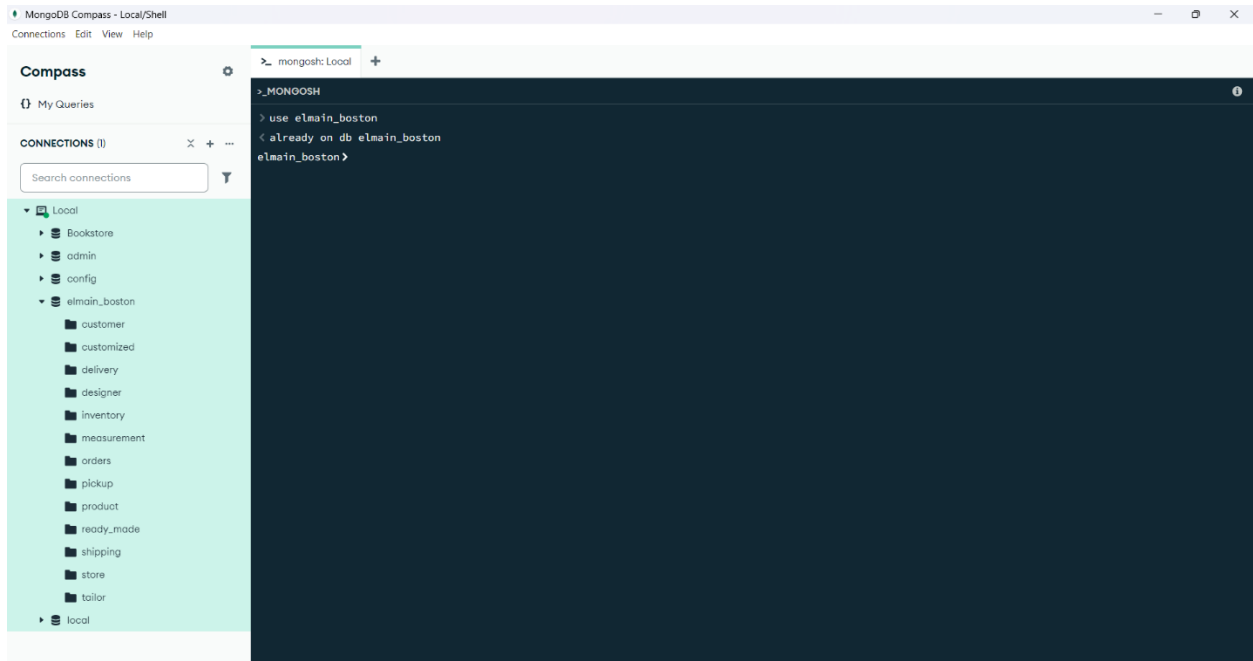
## The Elmain\_Boston database is hosted and operating in MongoDB platform



The screenshot shows the MongoDB Compass interface with the 'elmain\_boston' database selected. The left sidebar lists the database's collections: customer, customized, delivery, designer, inventory, measurement, orders, pickup, product, ready\_made, shipping, store, and tailor. The main panel displays a summary for five collections:

Collection Name	Storage size	Documents	Avg. document size	Indexes	Total index size
customer	40.96 kB	200	263.00 B	1	20.48 kB
customized	20.48 kB	100	113.00 B	1	20.48 kB
delivery	24.58 kB	200	134.00 B	1	20.48 kB
designer	20.48 kB	50	141.00 B	1	20.48 kB
inventory	20.48 kB	120	77.00 B	1	20.48 kB

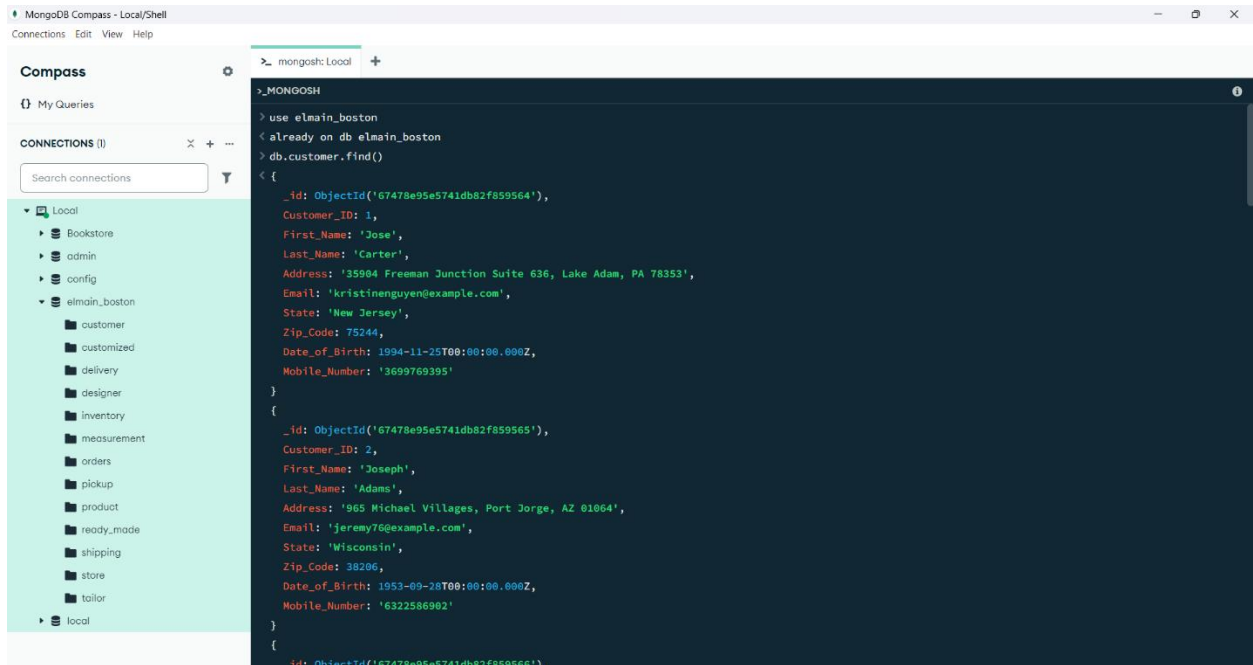
## Command to use the elmain\_boston database



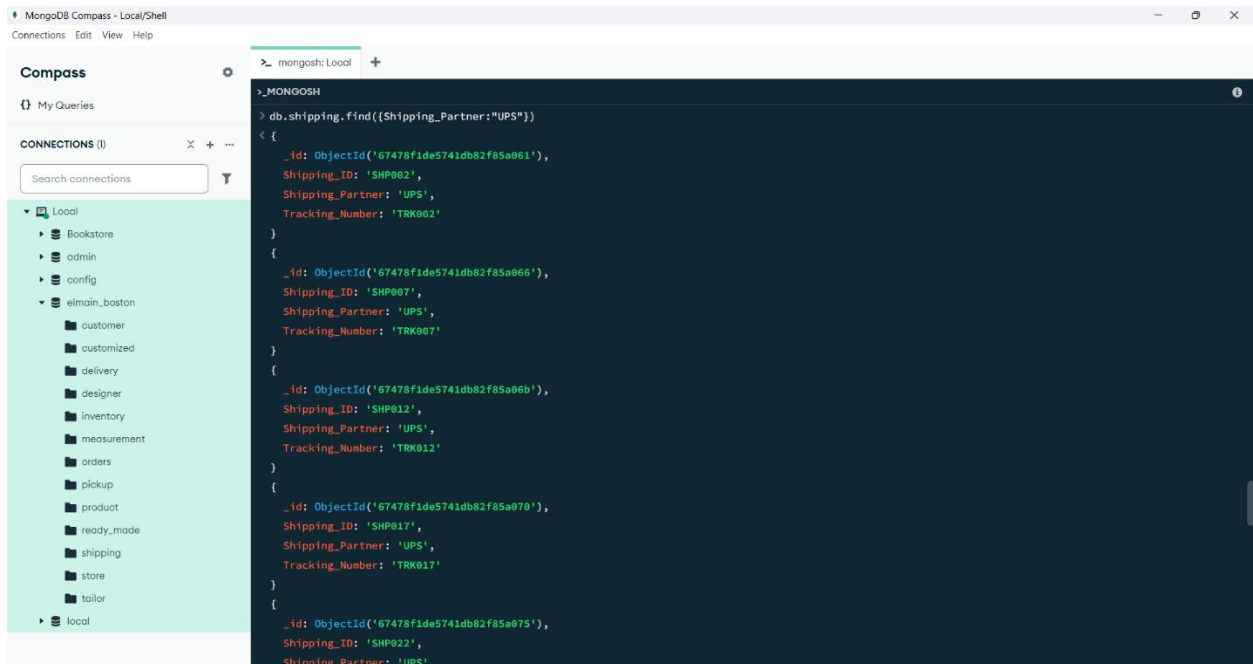
The screenshot shows the MongoDB Compass interface with the 'elmain\_boston' database selected. The left sidebar lists the database's collections. The main panel displays the MongoDB shell command to use the 'elmain\_boston' database:

```
> use elmain_boston
< already on db elmain_boston
elmain_boston>
```

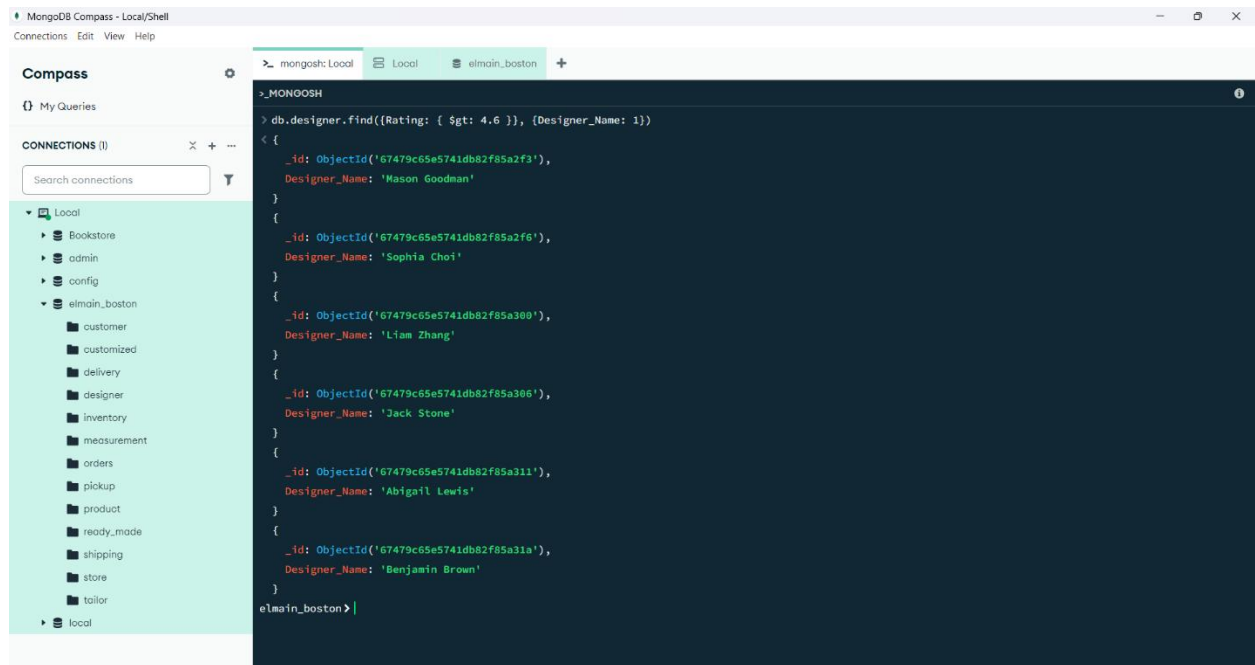
## Query to display all documents in the Customer Collection



## Query to display all documents from Shipping collection which has Shipping Partner as 'UPS'



## Query to display the Name of the Designers who have rating more than 4.6 in Designer Collection



## Query to find the average quantity of each product in all stores in the Inventory collection

