НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Інститут прикладного системного аналізу

ЛАБОРАТОРНА РОБОТА №3

з дисципліни «Високопродуктивні розподілені обчислювальні системи»

на тему: «Building MongoDB Environment»

Варіант №9

Виконали:
Студенти групи КІ-31мп
Гордун М. В.
Заіка Б. Ю.
Решетник О. О.

Перевірив:

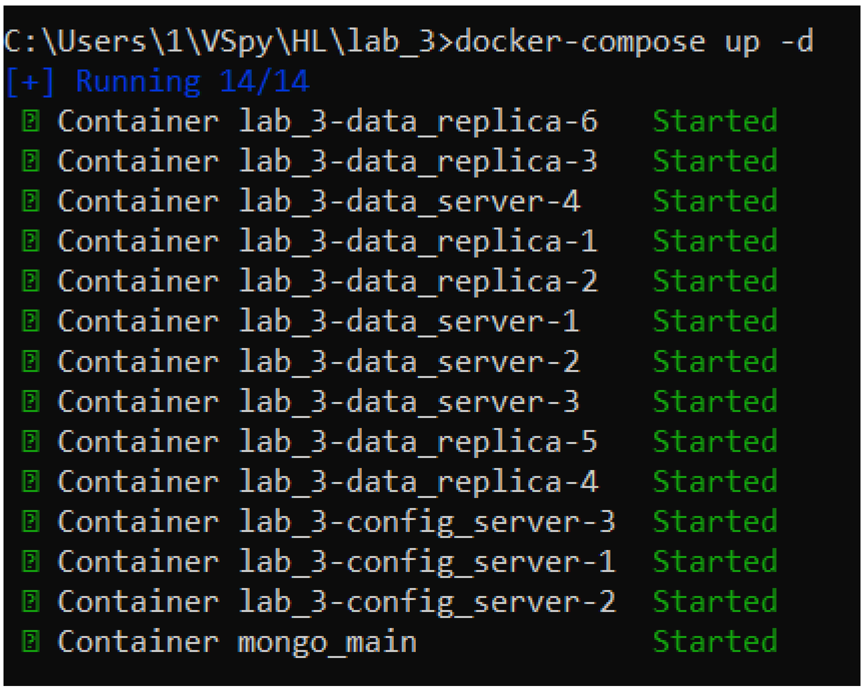
Кухарєв С. О.

Київ – 2023

Task 1. MongoDB sharding

docker-compose.yaml

```yaml
version: '3.8'

services:
  mongo_main:
    image: mongo
    container_name: mongo_main
    ports:
      - "27017:27017"
    networks:
      - mongo_net
    depends_on:
      - config_server
      - data_server
      - data_replica
    environment:
      - MONGO_INITDB_ROOT_USERNAME=admin
      - MONGO_INITDB_ROOT_PASSWORD=admin123

  config_server:
    image: mongo
    command: mongod --configsvr --replSet configReplSet
    networks:
      - mongo_net
    deploy:
      replicas: 3

  data_server:
    image: mongo
    command: mongod --shardsvr --replSet dataSet
    networks:
      - mongo_net
    deploy:
      replicas: 4

  data_replica:
    image: mongo
    command: mongod --shardsvr --replSet dataReplSet
    networks:
      - mongo_net
    deploy:
      replicas: 6

networks:
  mongo_net:
    driver: bridge
```



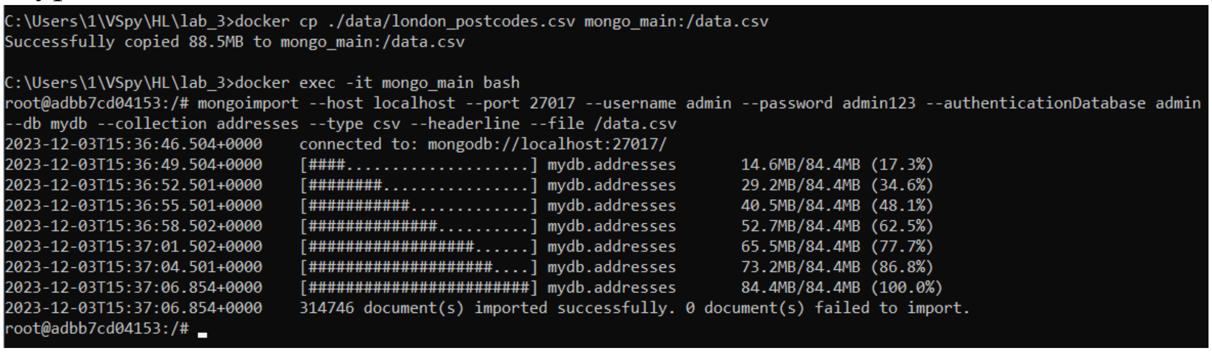
```
C:\Users\1\VSpy\HL\lab_3>docker-compose up -d
[+] Running 14/14
 ✔ Container lab_3-data_replica-6    Started
 ✔ Container lab_3-data_replica-3    Started
 ✔ Container lab_3-data_server-4     Started
 ✔ Container lab_3-data_replica-1    Started
 ✔ Container lab_3-data_replica-2    Started
 ✔ Container lab_3-data_server-1     Started
 ✔ Container lab_3-data_server-2     Started
 ✔ Container lab_3-data_server-3     Started
 ✔ Container lab_3-data_replica-5    Started
 ✔ Container lab_3-data_replica-4    Started
 ✔ Container lab_3-config_server-3   Started
 ✔ Container lab_3-config_server-1   Started
 ✔ Container lab_3-config_server-2   Started
 ✔ Container mongo_main              Started
```

Task 2. Import and balance data

docker cp ./data/london_postcodes.csv mongo_main:/data.csv

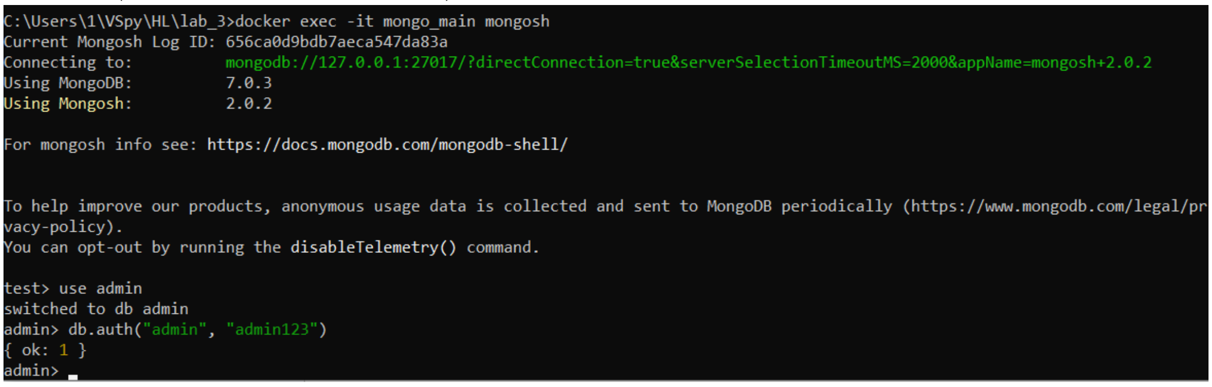
docker exec -it mongo_main bash

mongoimport --host localhost --port 27017 --username admin --password admin123 --authenticationDatabase admin --db mydb --collection addresses --type csv --headerline --file /data.csv

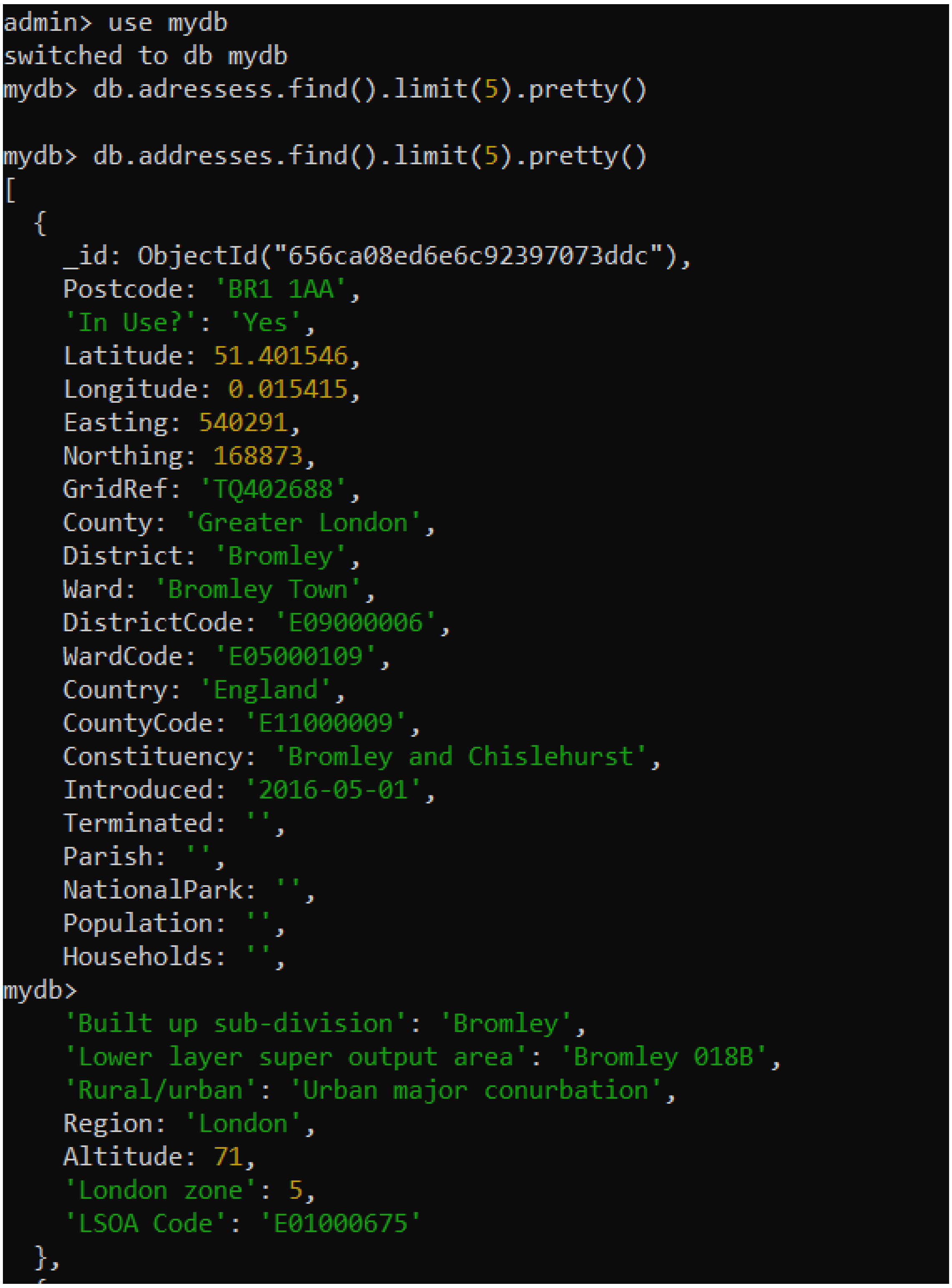


```
C:\Users\1\VSpy\HL\lab_3>docker cp ./data/london_postcodes.csv mongo_main:/data.csv
Successfully copied 88.5MB to mongo_main:/data.csv

C:\Users\1\VSpy\HL\lab_3>docker exec -it mongo_main bash
root@adbb7cd04153:/# mongoimport --host localhost --port 27017 --username admin --password admin123 --authenticationDatabase admin
--db mydb --collection addresses --type csv --headerline --file /data.csv
2023-12-03T15:36:46.504+0000    connected to: mongodb://localhost:27017/
2023-12-03T15:36:49.504+0000    [####....................] mydb.addresses       14.6MB/84.4MB (17.3%)
2023-12-03T15:36:52.501+0000    [########................] mydb.addresses       29.2MB/84.4MB (34.6%)
2023-12-03T15:36:55.501+0000    [###########.............] mydb.addresses       40.5MB/84.4MB (48.1%)
2023-12-03T15:36:58.502+0000    [##############..........] mydb.addresses       52.7MB/84.4MB (62.5%)
2023-12-03T15:37:01.502+0000    [##################......] mydb.addresses       65.5MB/84.4MB (77.7%)
2023-12-03T15:37:04.501+0000    [####################....] mydb.addresses       73.2MB/84.4MB (86.8%)
2023-12-03T15:37:06.854+0000    [########################] mydb.addresses       84.4MB/84.4MB (100.0%)
2023-12-03T15:37:06.854+0000    314746 document(s) imported successfully. 0 document(s) failed to import.
root@adbb7cd04153:/#
```

docker exec -it mongo_main mongosh
use admin
db.auth("admin", "admin123")



```
C:\Users\1\VSpy\HL\lab_3>docker exec -it mongo_main mongosh
Current Mongosh Log ID: 656ca0d9bdb7aeca547da83a
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.2
Using MongoDB:          7.0.3
Using Mongosh:          2.0.2

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


To help improve our products, anonymous usage data is collected and sent to MongoDB periodically (https://www.mongodb.com/legal/pr
vacy-policy).
You can opt-out by running the disableTelemetry() command.

test> use admin
switched to db admin
admin> db.auth("admin", "admin123")
{ ok: 1 }
admin>
```

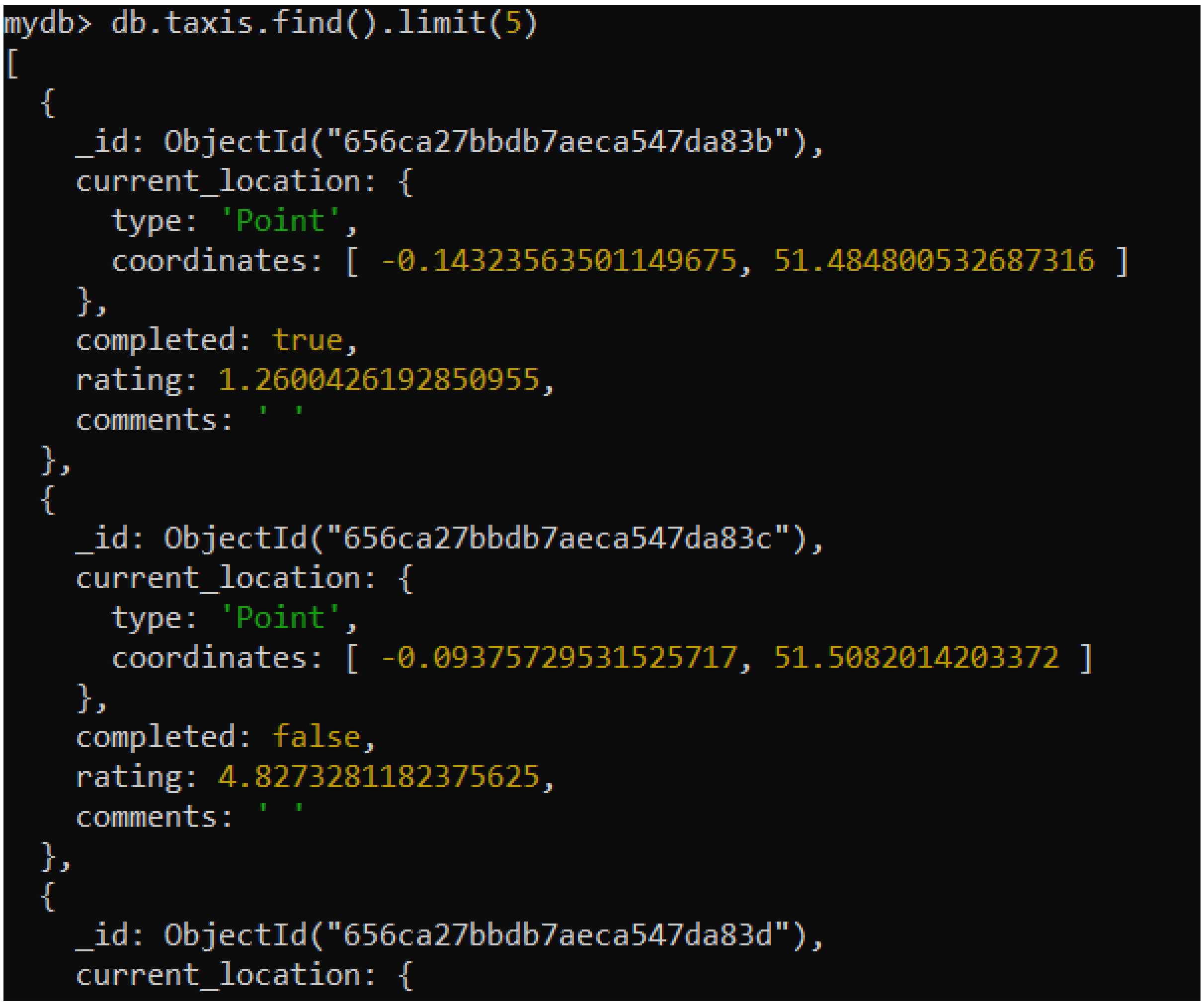


```
admin> use mydb
switched to db mydb
mydb> db.adresses.find().limit(5).pretty()

mydb> db.addresses.find().limit(5).pretty()
[
  {
    _id: ObjectId("656ca08ed6e6c92397073ddc"),
    Postcode: 'BR1 1AA',
    'In Use?': 'Yes',
    Latitude: 51.401546,
    Longitude: 0.015415,
    Easting: 540291,
    Northing: 168873,
    GridRef: 'TQ402688',
    County: 'Greater London',
    District: 'Bromley',
    Ward: 'Bromley Town',
    DistrictCode: 'E09000006',
    WardCode: 'E05000109',
    Country: 'England',
    CountyCode: 'E11000009',
    Constituency: 'Bromley and Chislehurst',
    Introduced: '2016-05-01',
    Terminated: '',
    Parish: '',
    NationalPark: '',
    Population: '',
    Households: '',
mydb>
    'Built up sub-division': 'Bromley',
    'Lower layer super output area': 'Bromley 018B',
    'Rural/urban': 'Urban major conurbation',
    Region: 'London',
    Altitude: 71,
    'London zone': 5,
    'LSOA Code': 'E01000675'
  },
```

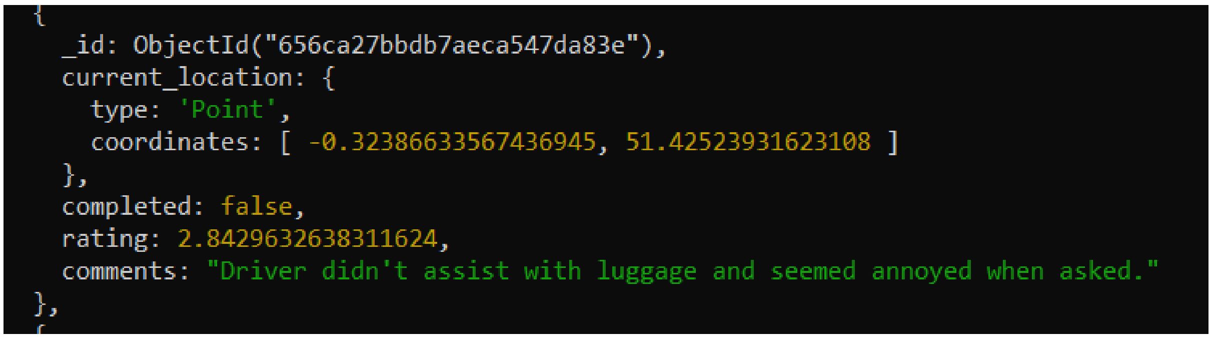
Task 3. Generate more data

db.createCollection("taxis")

```
function generateRandomComment(rating) {
  const randomIndex0 = Math.floor(Math.random() * commentsListPostive.length);
          const randomIndex1 = Math.floor(Math.random() *
commentsListNegative.length);
  if (Math.random() > 0.6) {
    if (rating >= 3) {
        return commentsListPostive[randomIndex0];
    } else {
        return commentsListNegative[randomIndex1];
    }
    }
  else {
    return " ";
  }
}

const addresses = db.addresses.find({}, { _id: 0, Latitude: 1, Longitude: 1
}).toArray();
for (let i = 0; i < 300; i++) {
  var randon_address = addresses[Math.random()*addresses.length>>0]
   var random_shift = (Math.random() < 0.5 ? -1 : 1) * Math.random() * (0.02 -
(0.002)) + (0.002)
  var random_shift1 = (Math.random() < 0.5 ? -1 : 1) * Math.random() * (0.02 -
(0.002)) + (0.002)
  var random_rating = Math.random() * (5 - (1)) + (1)

const driver = {
    current_location: {
      type: "Point",
            coordinates: [randon_address.Longitude + random_shift,
randon_address.Latitude + random_shift1] // Use random coordinates for
dropoff
    },
    completed: Math.random() > 0.5,
    rating: random_rating,
    comments: generateRandomComment(random_rating)
  };
  db.taxis.insertOne(driver);
}
```

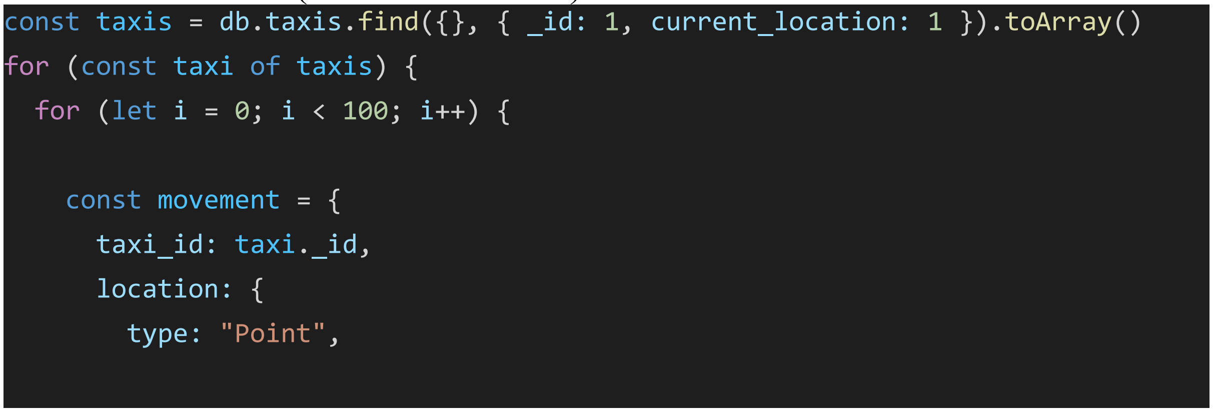


```
mydb> db.taxis.find().limit(5)
[
  {
    _id: ObjectId("656ca27bbdb7aeca547da83b"),
    current_location: {
      type: 'Point',
      coordinates: [ -0.14323563501149675, 51.484800532687316 ]
    },
    completed: true,
    rating: 1.2600426192850955,
    comments: ' '
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da83c"),
    current_location: {
      type: 'Point',
      coordinates: [ -0.09375729531525717, 51.5082014203372 ]
    },
    completed: false,
    rating: 4.8273281182375625,
    comments: ' '
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da83d"),
    current_location: {
```

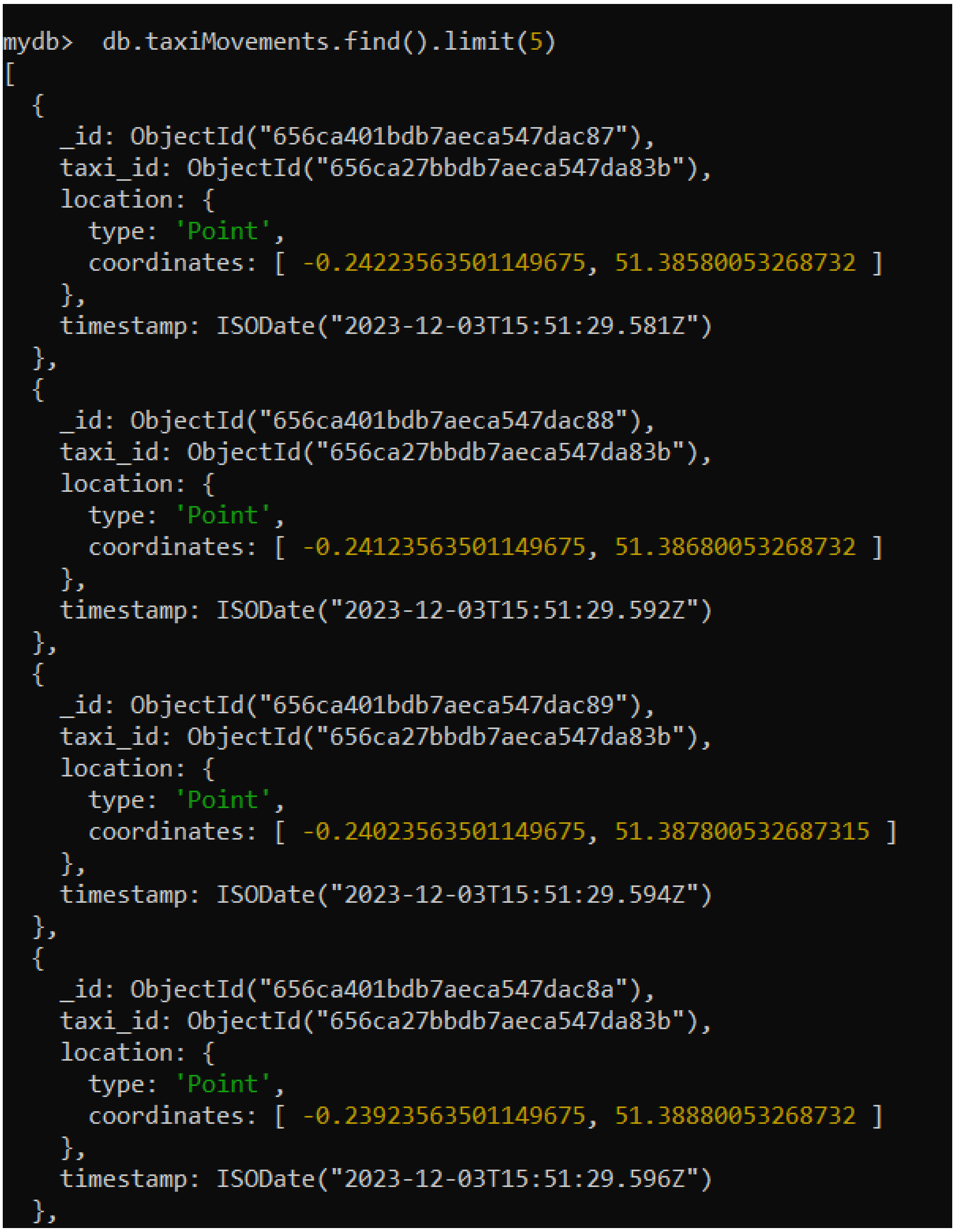


```
  {
    _id: ObjectId("656ca27bbdb7aeca547da83e"),
    current_location: {
      type: 'Point',
      coordinates: [ -0.32386633567436945, 51.42523931623108 ]
    },
    completed: false,
    rating: 2.8429632638311624,
    comments: "Driver didn't assist with luggage and seemed annoyed when asked."
  },
```

db.createCollection("taxiMovements")



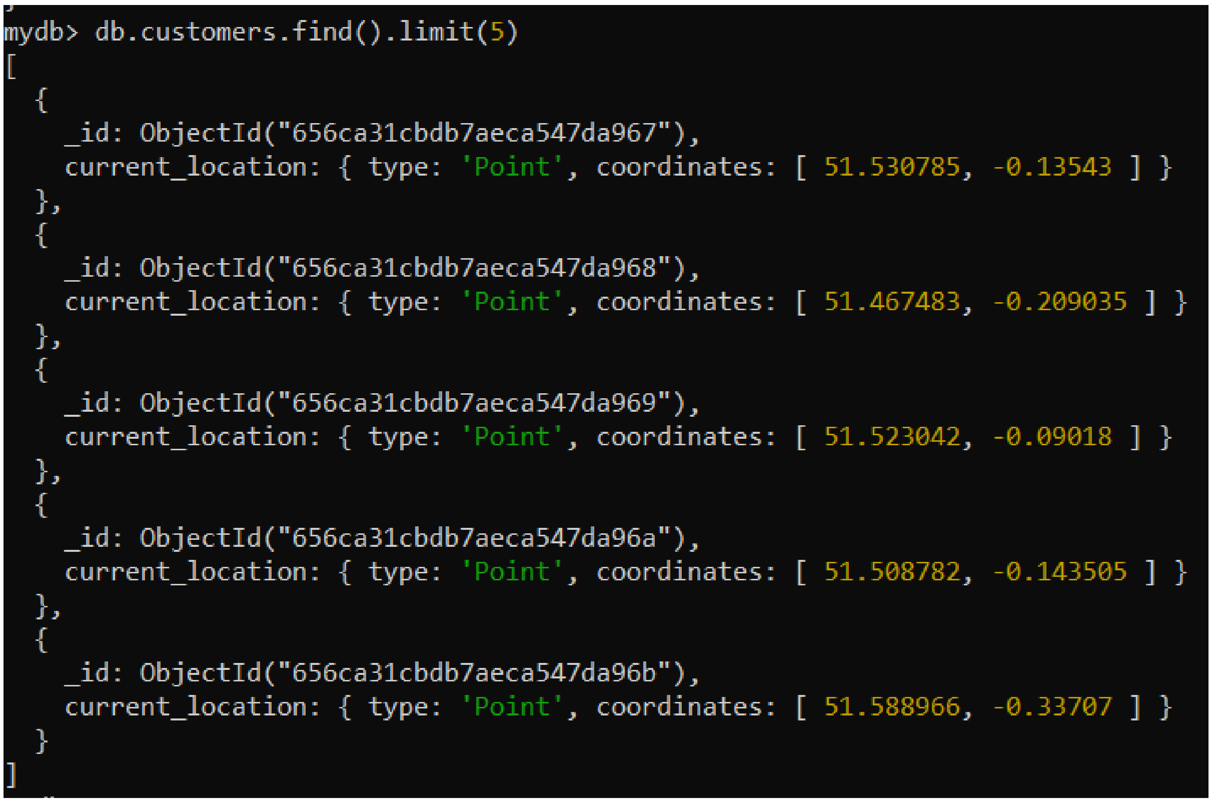
```
const taxis = db.taxis.find({}, { _id: 1, current_location: 1 }).toArray()
for (const taxi of taxis) {
  for (let i = 0; i < 100; i++) {

    const movement = {
      taxi_id: taxi._id,
      location: {
        type: "Point",
        coordinates: [taxi.current_location.coordinates[0] - (99 - i) * 0.001, 
taxi.current_location.coordinates[1] - (99 - i) * 0.001]
      },
      timestamp: new Date()
    };
    db.taxiMovements.insertOne(movement);
  }
}
```



```
mydb>  db.taxiMovements.find().limit(5)
[
  {
    _id: ObjectId("656ca401bdb7aeca547dac87"),
    taxi_id: ObjectId("656ca27bbdb7aeca547da83b"),
    location: {
      type: 'Point',
      coordinates: [ -0.24223563501149675, 51.38580053268732 ]
    },
    timestamp: ISODate("2023-12-03T15:51:29.581Z")
  },
  {
    _id: ObjectId("656ca401bdb7aeca547dac88"),
    taxi_id: ObjectId("656ca27bbdb7aeca547da83b"),
    location: {
      type: 'Point',
      coordinates: [ -0.24123563501149675, 51.38680053268732 ]
    },
    timestamp: ISODate("2023-12-03T15:51:29.592Z")
  },
  {
    _id: ObjectId("656ca401bdb7aeca547dac89"),
    taxi_id: ObjectId("656ca27bbdb7aeca547da83b"),
    location: {
      type: 'Point',
      coordinates: [ -0.24023563501149675, 51.387800532687315 ]
    },
    timestamp: ISODate("2023-12-03T15:51:29.594Z")
  },
  {
    _id: ObjectId("656ca401bdb7aeca547dac8a"),
    taxi_id: ObjectId("656ca27bbdb7aeca547da83b"),
    location: {
      type: 'Point',
      coordinates: [ -0.23923563501149675, 51.38880053268732 ]
    },
    timestamp: ISODate("2023-12-03T15:51:29.596Z")
  },
```

db.createCollection("customers")

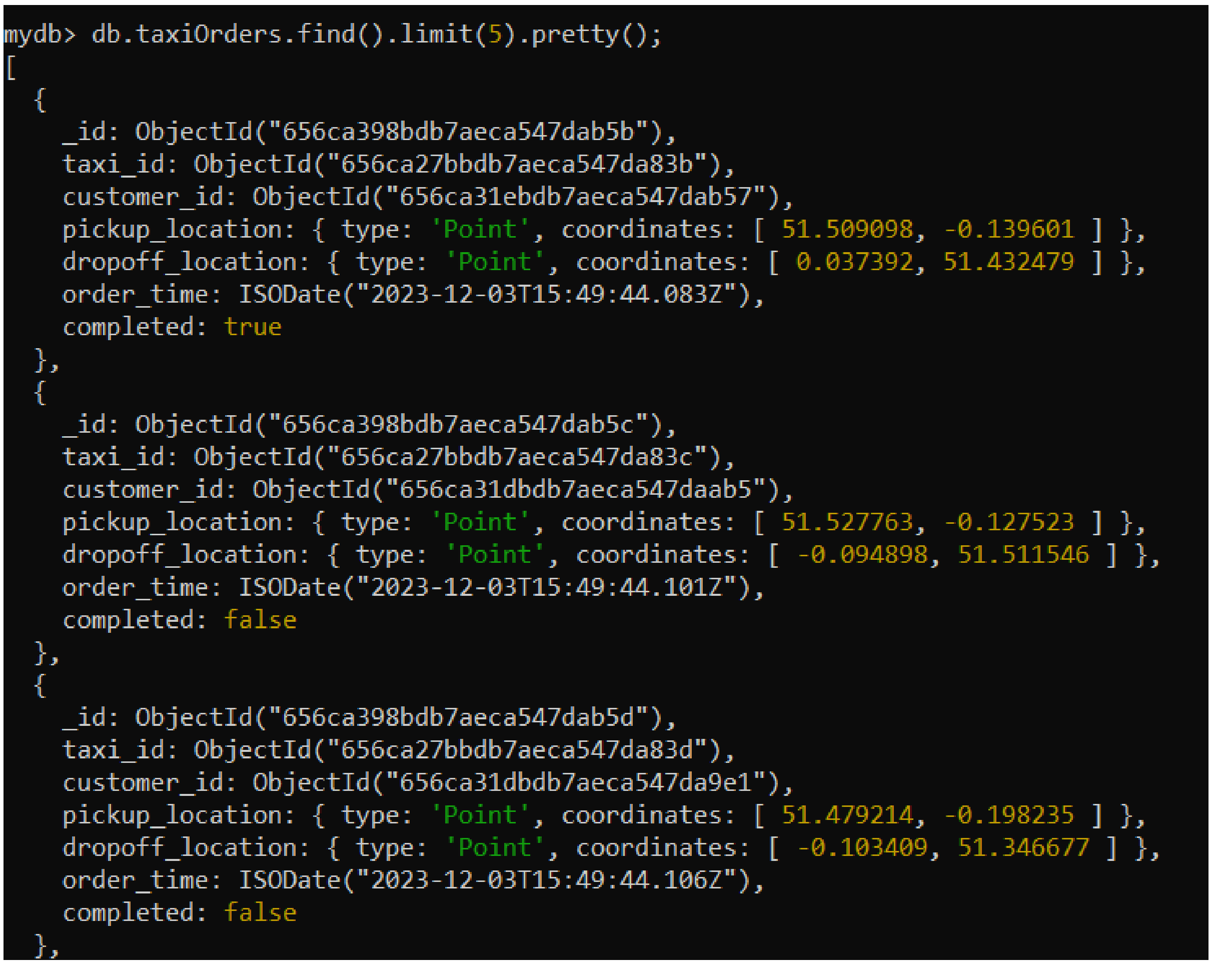
```
const addresses = db.addresses.find({}, { _id: 0, Latitude: 1, Longitude: 1
}).toArray();
for (let i = 0; i < 500; i++) {
  var random_address = addresses[Math.random()*addresses.length>>0]
const customer = {
    current_location: {
      type: "Point",
      coordinates: [random_address.Latitude, random_address.Longitude]  // Use
random coordinates for dropoff
    },
  };
  db.customers.insertOne(customer);
}
```



```
mydb> db.customers.find().limit(5)
[
  {
    _id: ObjectId("656ca31cbdb7aeca547da967"),
    current_location: { type: 'Point', coordinates: [ 51.530785, -0.13543 ] }
  },
  {
    _id: ObjectId("656ca31cbdb7aeca547da968"),
    current_location: { type: 'Point', coordinates: [ 51.467483, -0.209035 ] }
  },
  {
    _id: ObjectId("656ca31cbdb7aeca547da969"),
    current_location: { type: 'Point', coordinates: [ 51.523042, -0.09018 ] }
  },
  {
    _id: ObjectId("656ca31cbdb7aeca547da96a"),
    current_location: { type: 'Point', coordinates: [ 51.508782, -0.143505 ] }
  },
  {
    _id: ObjectId("656ca31cbdb7aeca547da96b"),
    current_location: { type: 'Point', coordinates: [ 51.588966, -0.33707 ] }
  }
]
```

db.createCollection("taxiOrders")

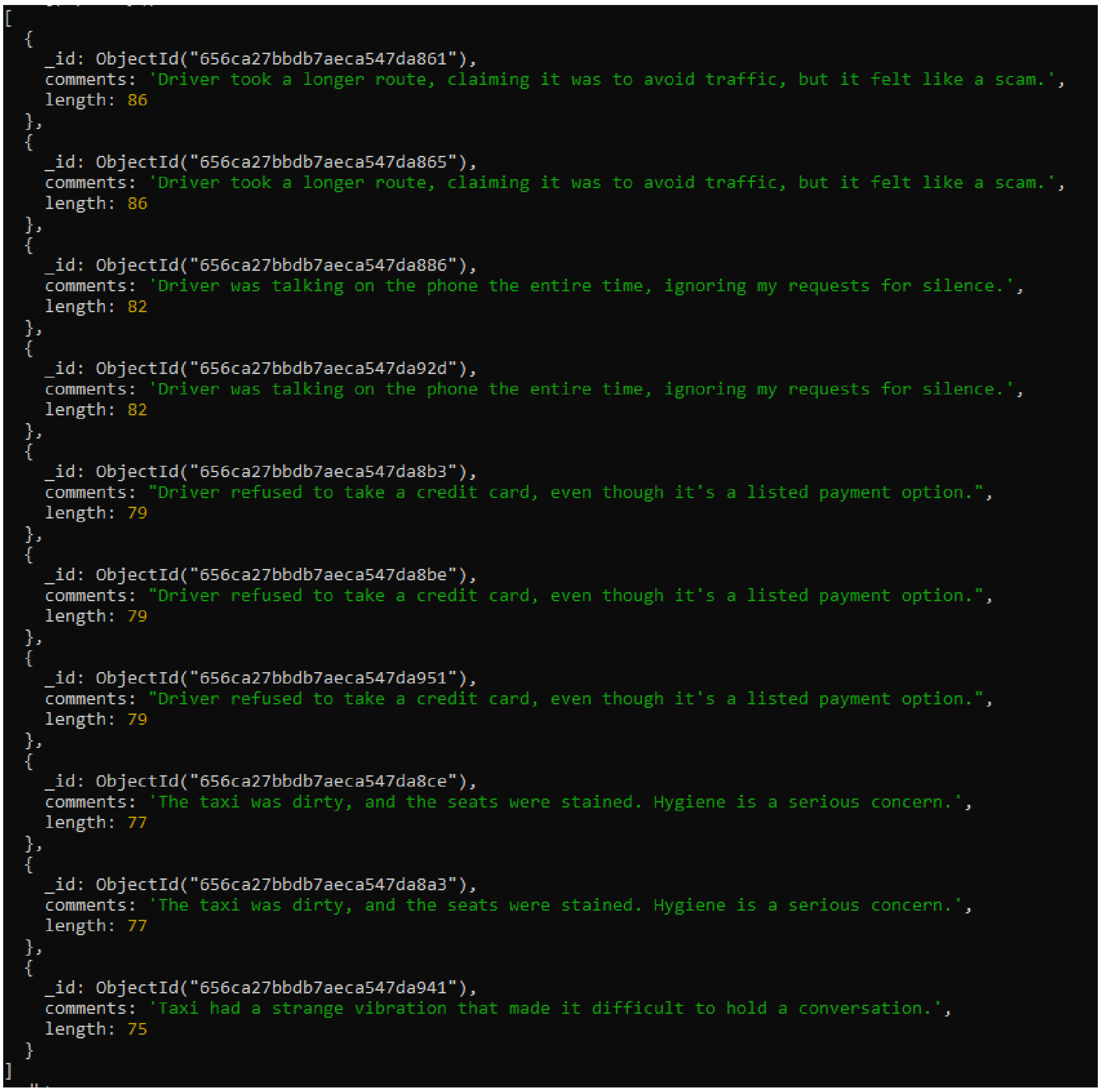
```
const  pickup_address  =  db.customers.find({},  {  _id:  1,  current_location:  1
}).toArray();
const taxis = db.taxis.find({}, { _id: 1, completed: 1 }).toArray()
for (const taxi of taxis) {
    var randon_address = addresses[Math.random()*addresses.length>>0]
                                        var          random_customer          =
pickup_address[Math.random()*pickup_address.length>>0]
    const order = {
      taxi_id: taxi._id,
      customer_id: random_customer._id,
      pickup_location: random_customer.current_location,
      dropoff_location: {
        type: "Point",
          coordinates: [randon_address.Longitude, randon_address.Latitude]  // Use random coordinates for dropoff
      },
      order_time: new Date(),
      completed: taxi.completed,
    };
    db.taxiOrders.insertOne(order);
}

// Test Taxi Orders
db.taxiOrders.find().limit(5).pretty();
```



```
mydb> db.taxiOrders.find().limit(5).pretty();
[
  {
    _id: ObjectId("656ca398bdb7aeca547dab5b"),
    taxi_id: ObjectId("656ca27bbdb7aeca547da83b"),
    customer_id: ObjectId("656ca31ebdb7aeca547dab57"),
    pickup_location: { type: 'Point', coordinates: [ 51.509098, -0.139601 ] },
    dropoff_location: { type: 'Point', coordinates: [ 0.037392, 51.432479 ] },
    order_time: ISODate("2023-12-03T15:49:44.083Z"),
    completed: true
  },
  {
    _id: ObjectId("656ca398bdb7aeca547dab5c"),
    taxi_id: ObjectId("656ca27bbdb7aeca547da83c"),
    customer_id: ObjectId("656ca31dbdb7aeca547daab5"),
    pickup_location: { type: 'Point', coordinates: [ 51.527763, -0.127523 ] },
    dropoff_location: { type: 'Point', coordinates: [ -0.094898, 51.511546 ] },
    order_time: ISODate("2023-12-03T15:49:44.101Z"),
    completed: false
  },
  {
    _id: ObjectId("656ca398bdb7aeca547dab5d"),
    taxi_id: ObjectId("656ca27bbdb7aeca547da83d"),
    customer_id: ObjectId("656ca31dbdb7aeca547da9e1"),
    pickup_location: { type: 'Point', coordinates: [ 51.479214, -0.198235 ] },
    dropoff_location: { type: 'Point', coordinates: [ -0.103409, 51.346677 ] },
    order_time: ISODate("2023-12-03T15:49:44.106Z"),
    completed: false
  },
```

Task 4.

```
db.taxis.aggregate([
  {
      $project: {
          length: { $strLenCP: "$comments" },
          comments: 1
      }
  },
  {
      $sort: { length: -1 }
  },
  {
      $limit: 10
  }
]).pretty();
```



```
[
  {
    _id: ObjectId("656ca27bbdb7aeca547da861"),
    comments: 'Driver took a longer route, claiming it was to avoid traffic, but it felt like a scam.',
    length: 86
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da865"),
    comments: 'Driver took a longer route, claiming it was to avoid traffic, but it felt like a scam.',
    length: 86
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da886"),
    comments: 'Driver was talking on the phone the entire time, ignoring my requests for silence.',
    length: 82
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da92d"),
    comments: 'Driver was talking on the phone the entire time, ignoring my requests for silence.',
    length: 82
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da8b3"),
    comments: "Driver refused to take a credit card, even though it's a listed payment option.",
    length: 79
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da8be"),
    comments: "Driver refused to take a credit card, even though it's a listed payment option.",
    length: 79
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da951"),
    comments: "Driver refused to take a credit card, even though it's a listed payment option.",
    length: 79
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da8ce"),
    comments: 'The taxi was dirty, and the seats were stained. Hygiene is a serious concern.',
    length: 77
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da8a3"),
    comments: 'The taxi was dirty, and the seats were stained. Hygiene is a serious concern.',
    length: 77
  },
  {
    _id: ObjectId("656ca27bbdb7aeca547da941"),
    comments: 'Taxi had a strange vibration that made it difficult to hold a conversation.',
    length: 75
  }
]
```