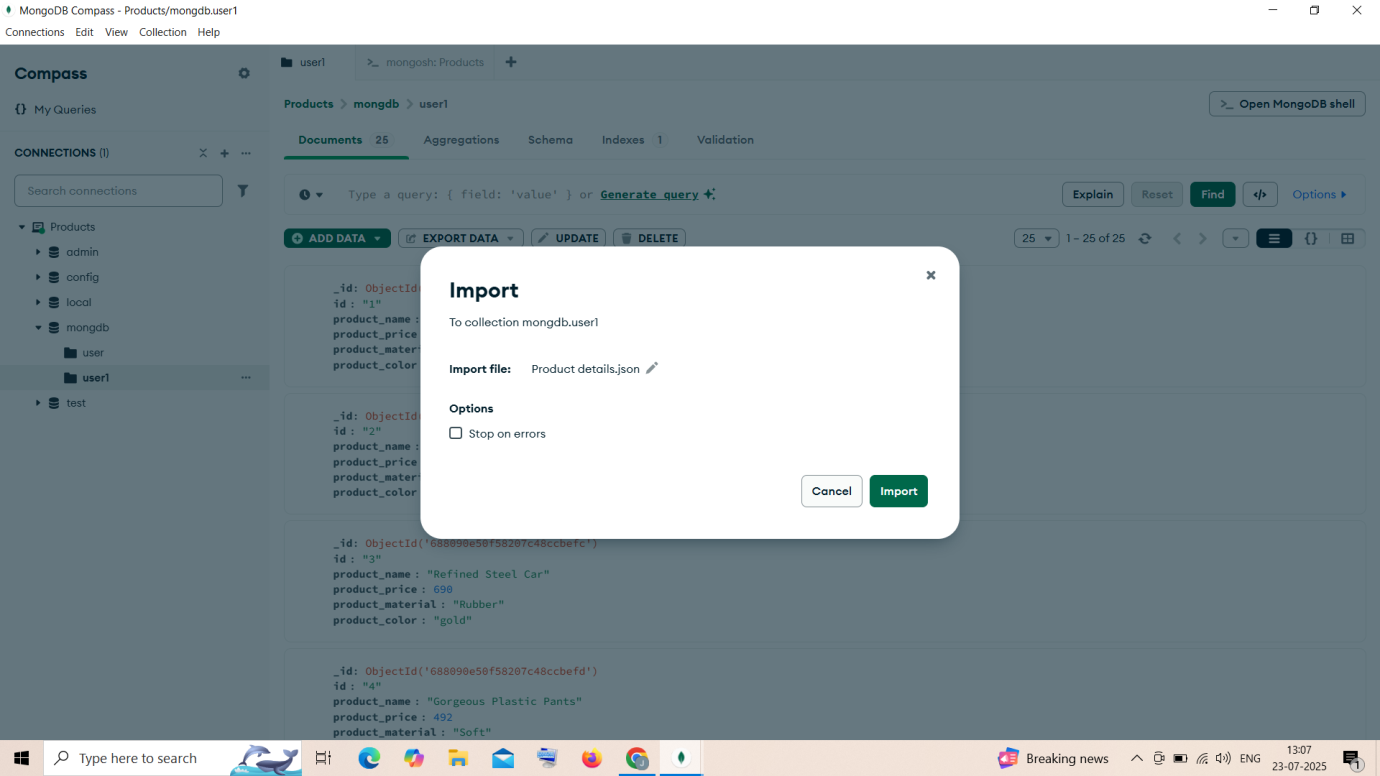
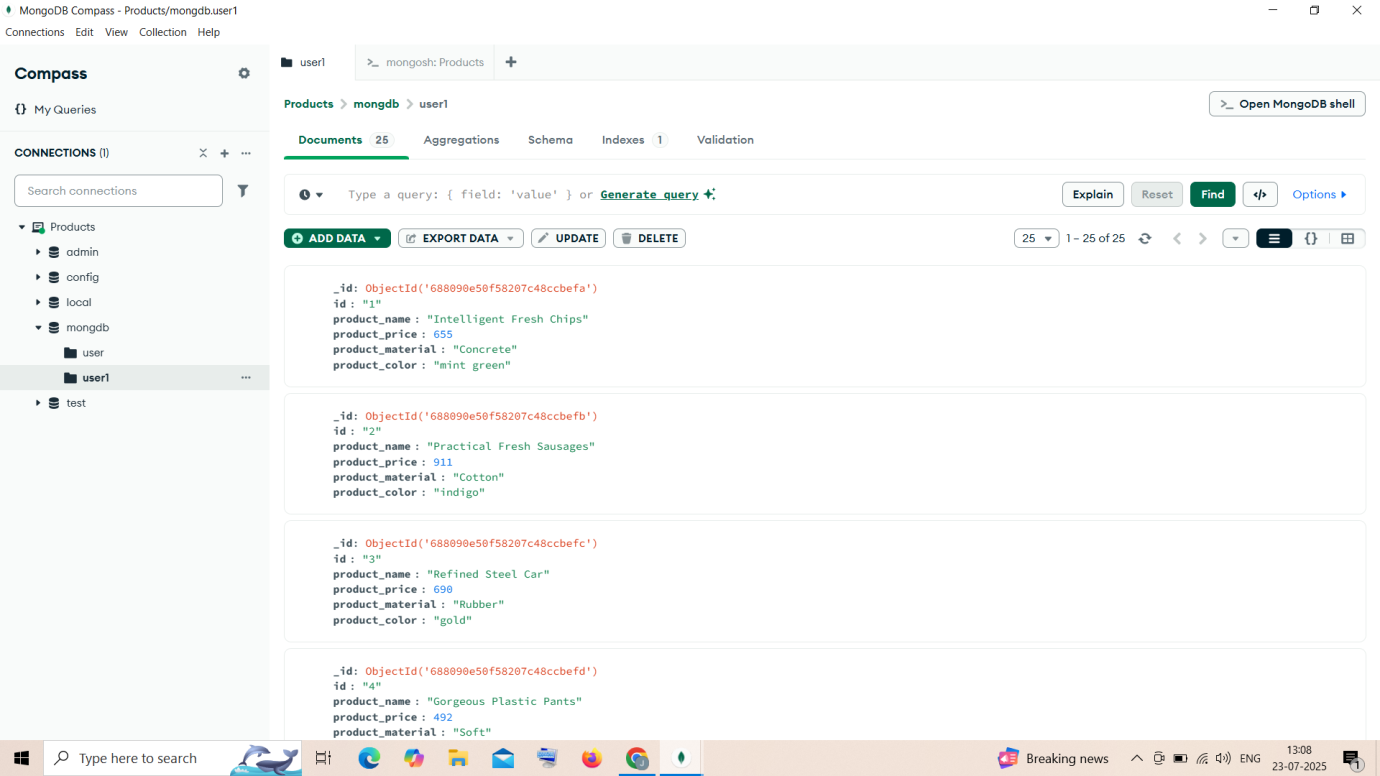
# DATA ENGINEERING – HEXAWARE- Job Aoushadan N

[Jobaoushadancse2021@jerusalemengg.ac.in](mailto:Jobaoushadancse2021@jerusalemengg.ac.in)

STEPS:

1.Import Json file from the file explorer



2. It can be viewed like this by using find

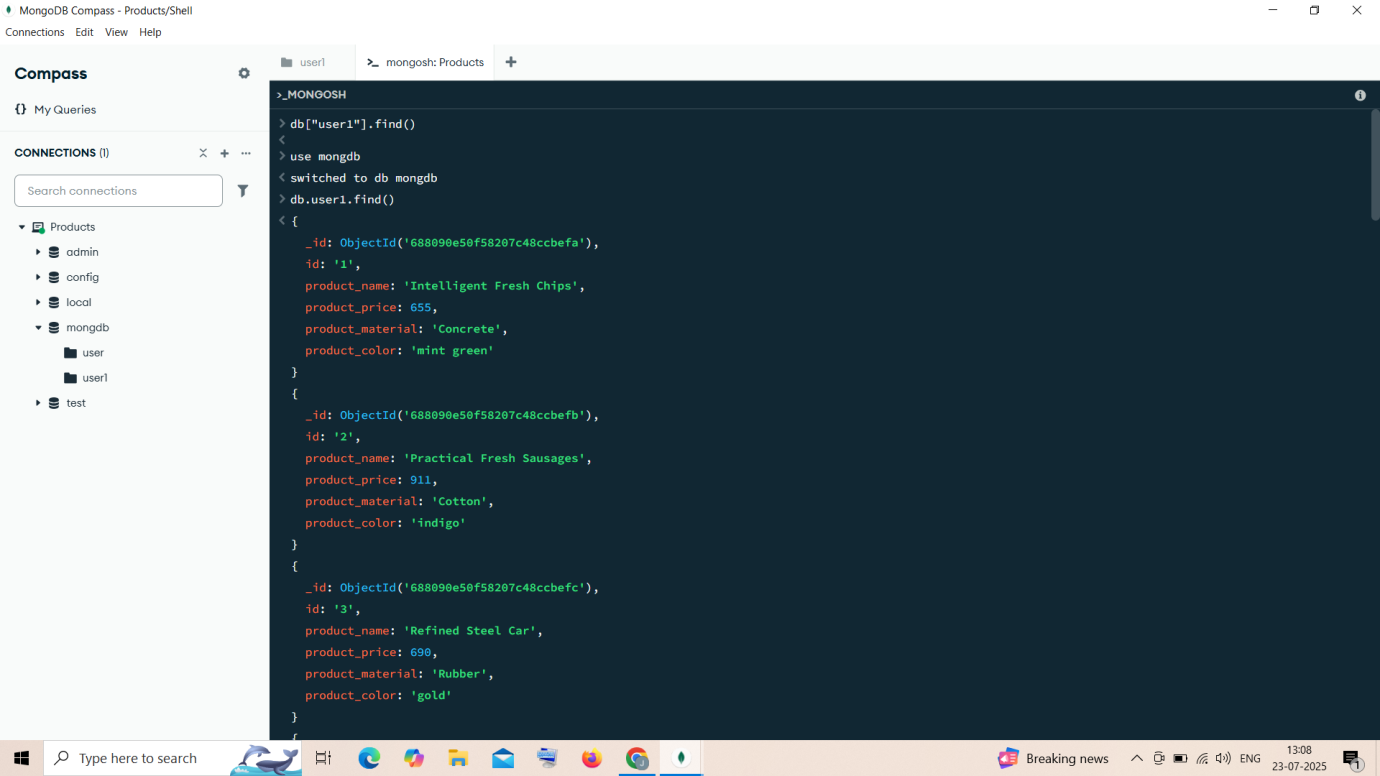
Database – mongdb

Collection – user1

QUERIES:

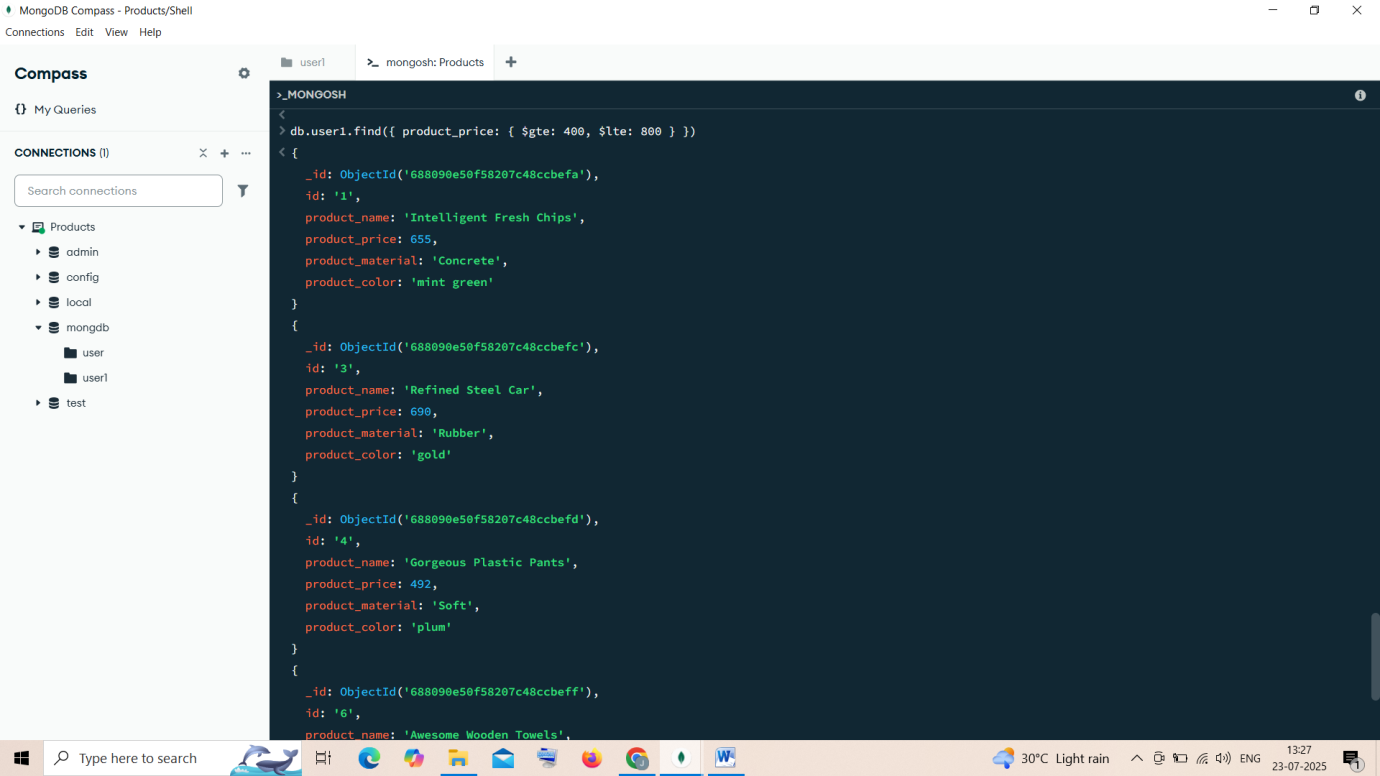
### 1. ****Find all the information about each product****

db.user1.find()



### 2. ****Find the product price which are between 400 to 800****

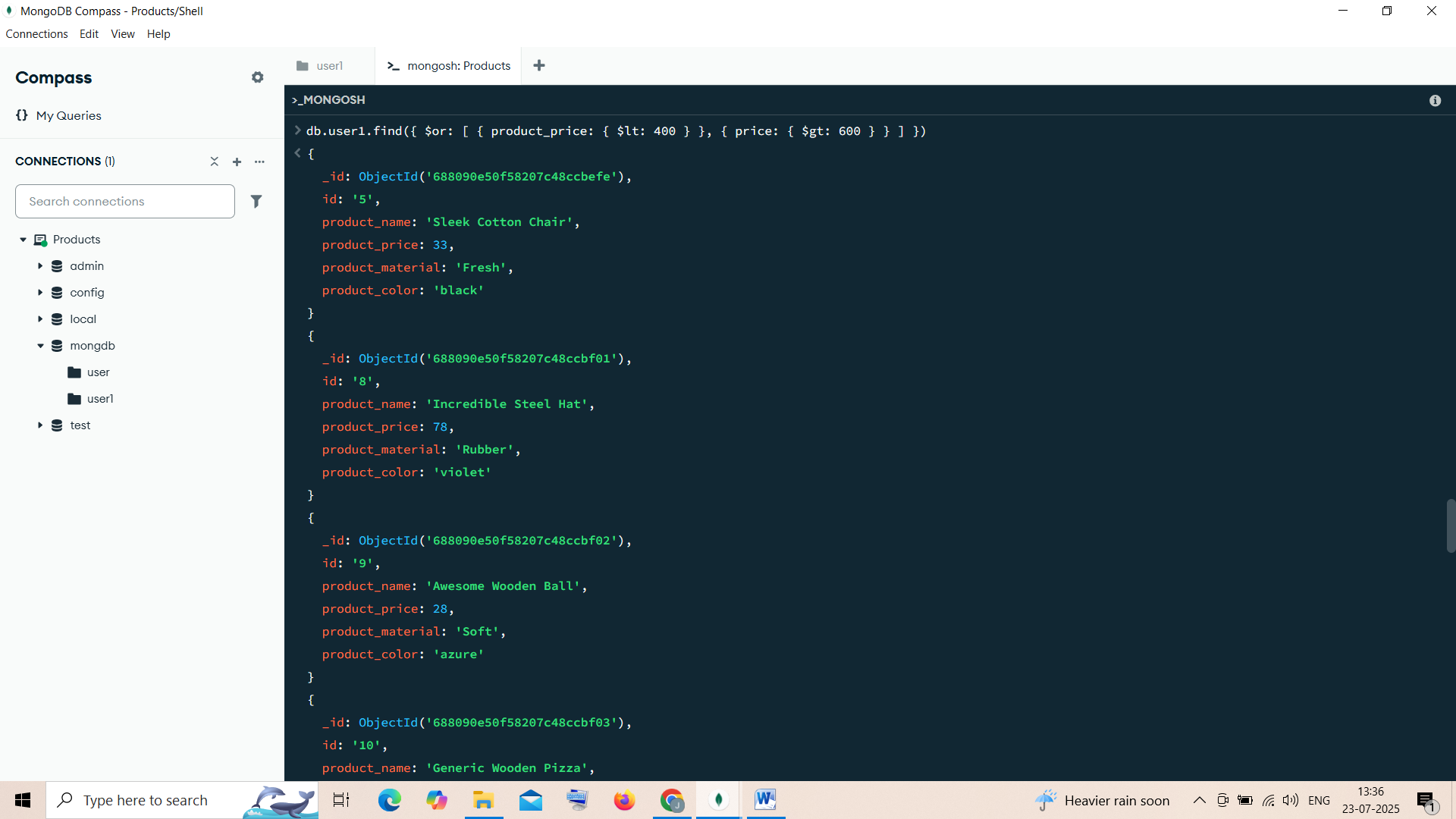
db.user1.find({ product\_price: { $gte: 400, $lte: 800 } })



### 3. ****Find the product price which are not between 400 to 600****

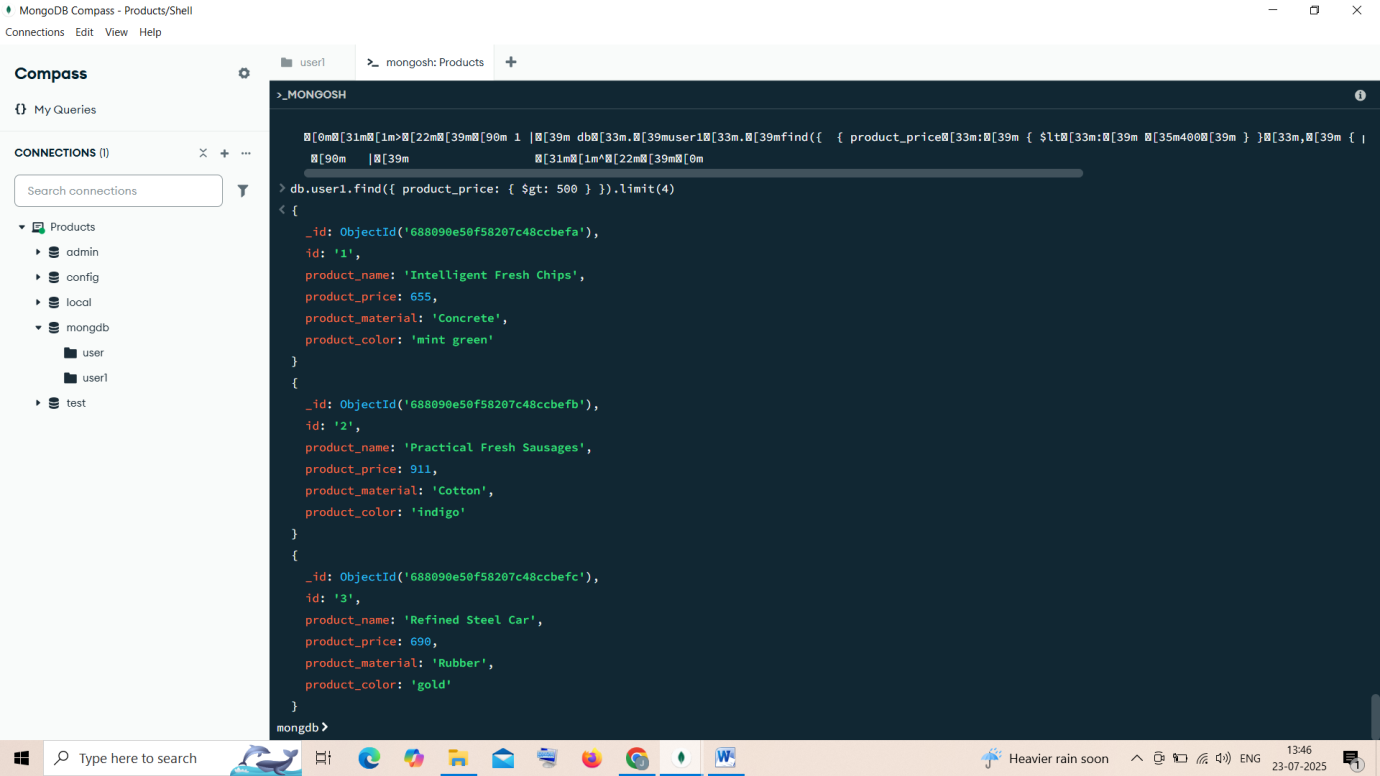
db.user1.find({ $or: [ { product\_price: { $lt: 400 } }, { price: { $gt: 600 } } ] })

$or – performs a logical OR operation on an array,selects the documents that satisfy atleast one of the <expressions>



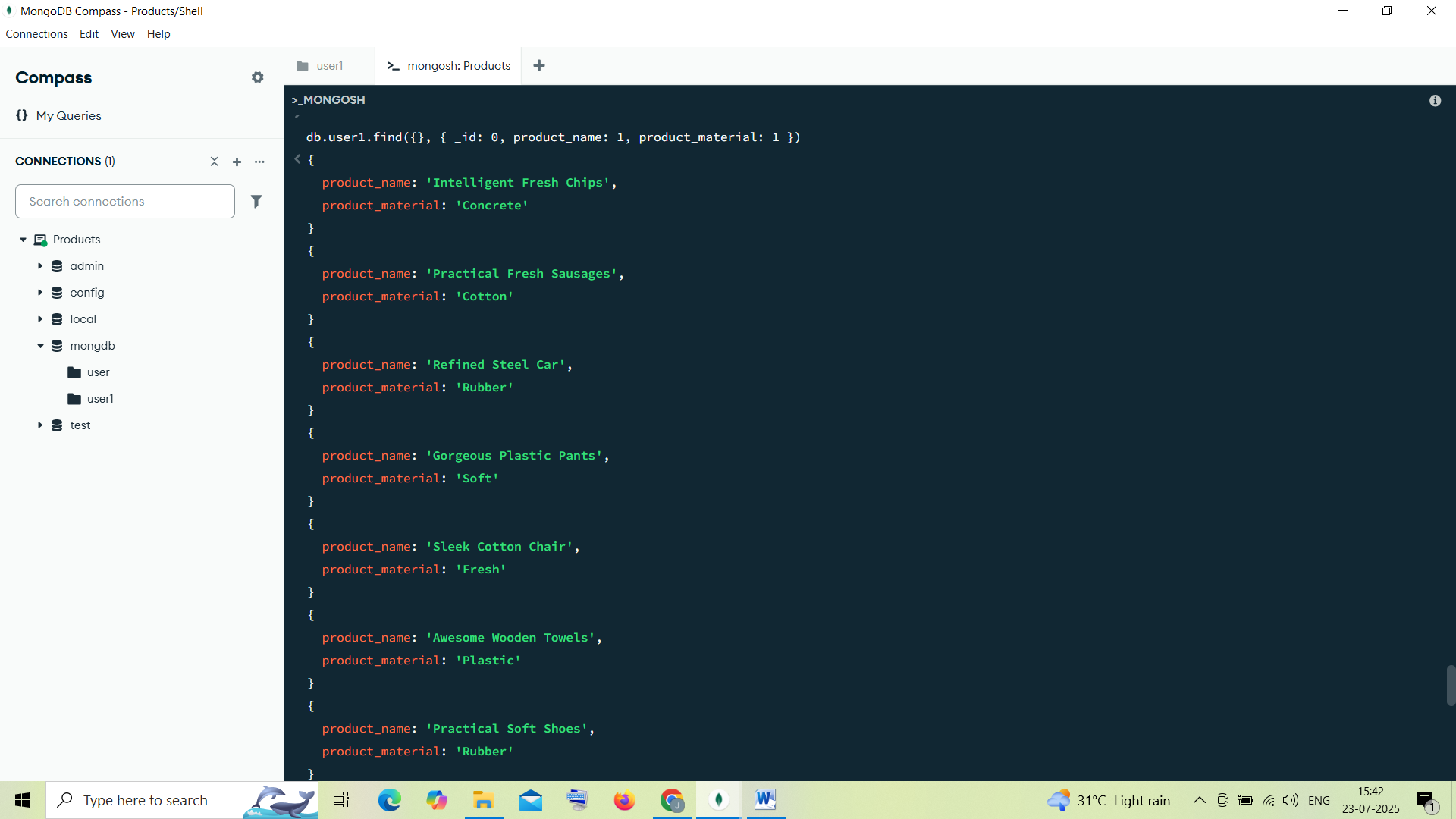
### 4. ****List the four products which are greater than 500 in price****

db.user1.find({ product\_price: { $gt: 500 } }).limit(4)



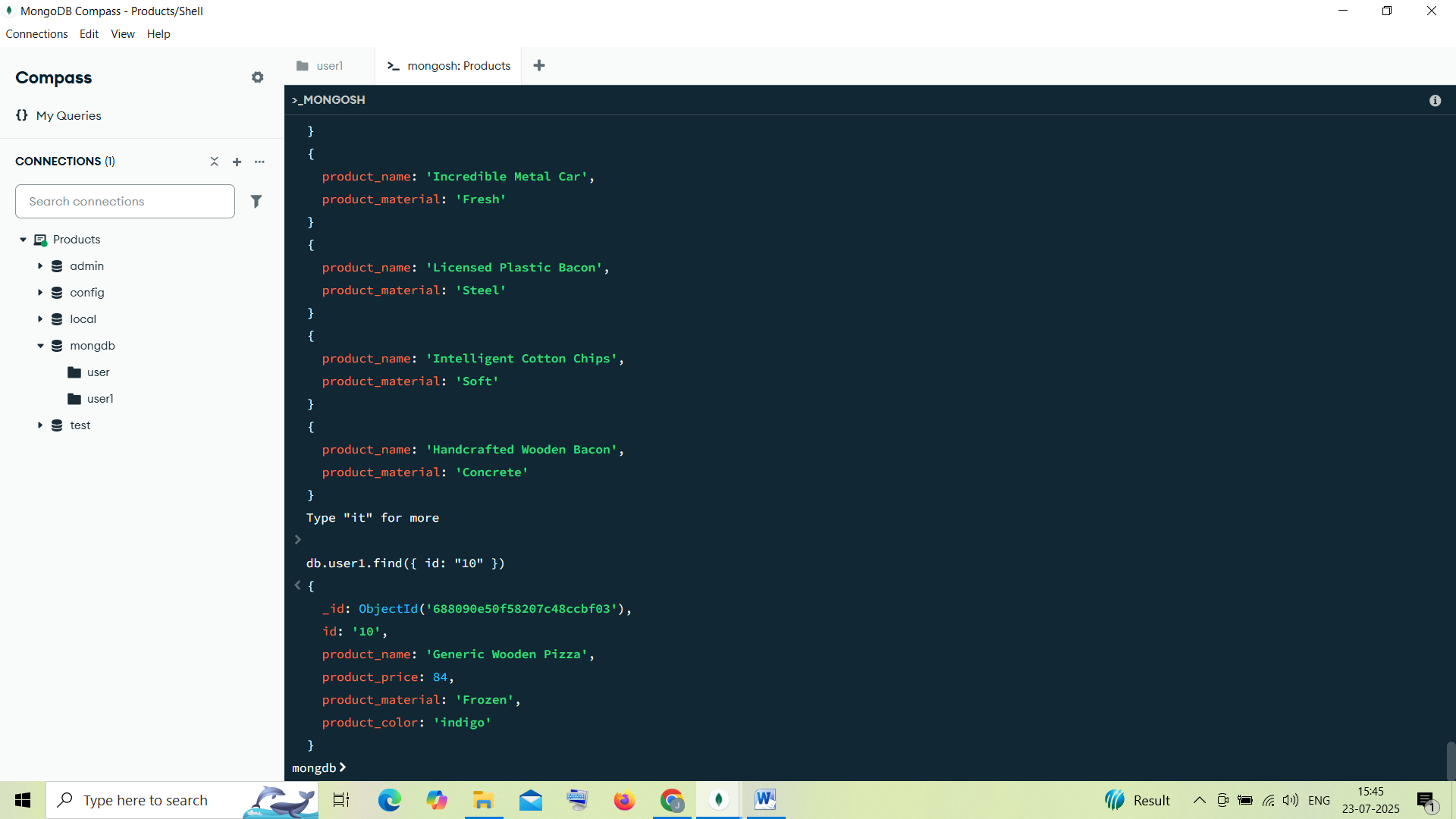
### 5. ****Find the product name and product material of each product****

db.user1.find({}, { \_id: 0, product\_name: 1, product\_material: 1 })



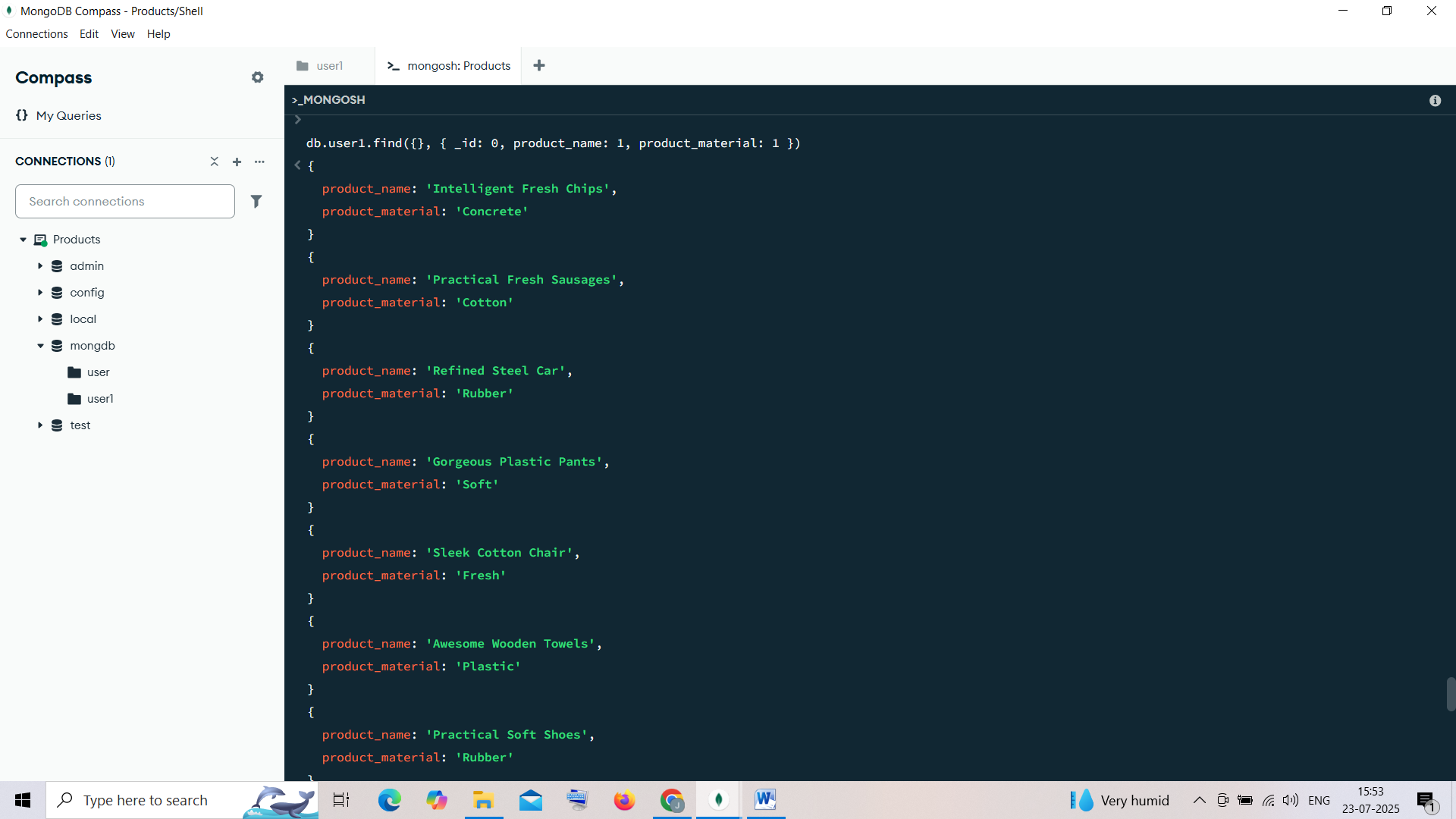
### 6. ****Find the product with a row id of 10****

db.user1.find({ id: "10" })



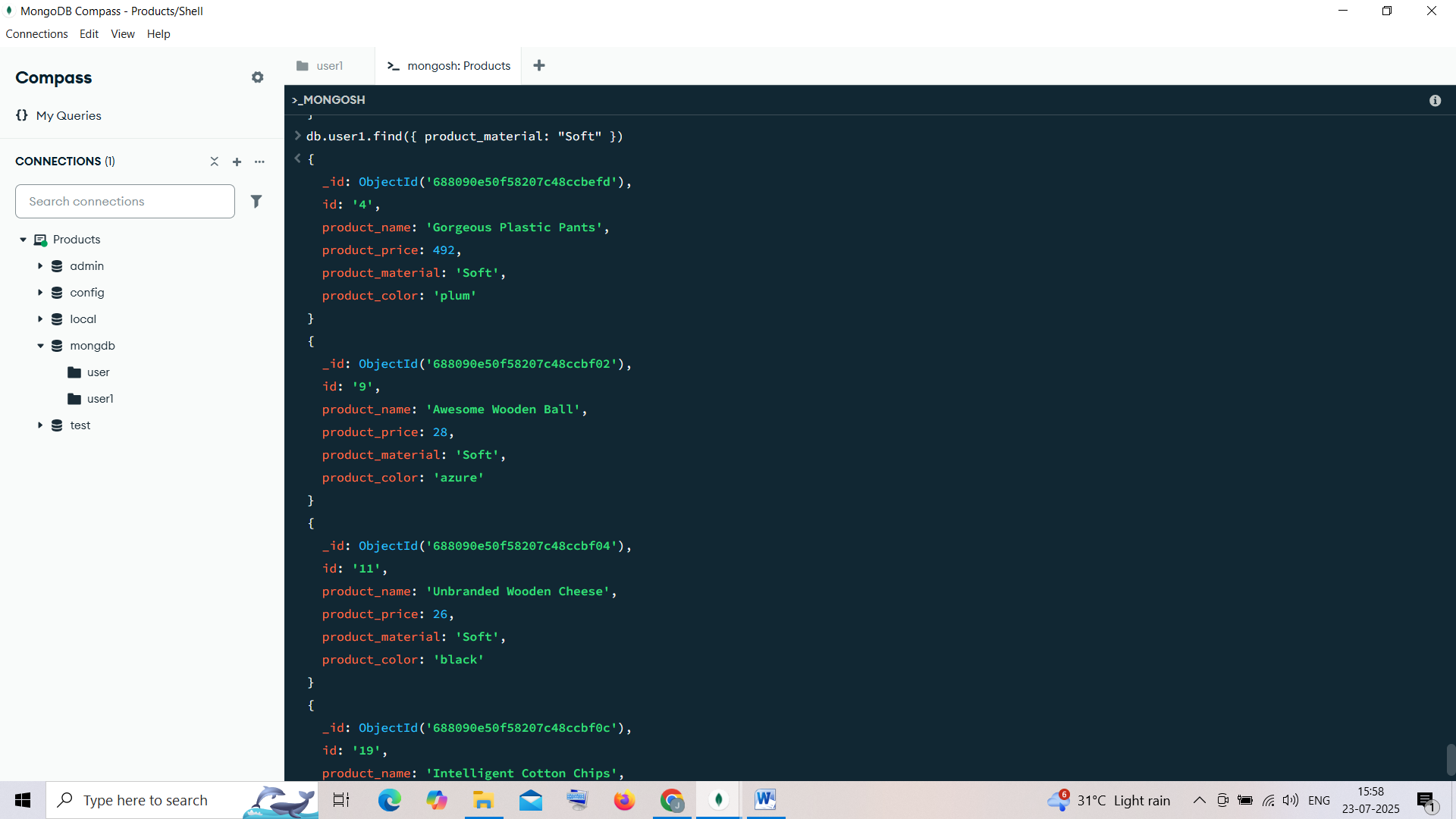
### 7. ****Find only the product name and product material****

db.user1.find({}, { \_id: 0, product\_name: 1, product\_material: 1 })



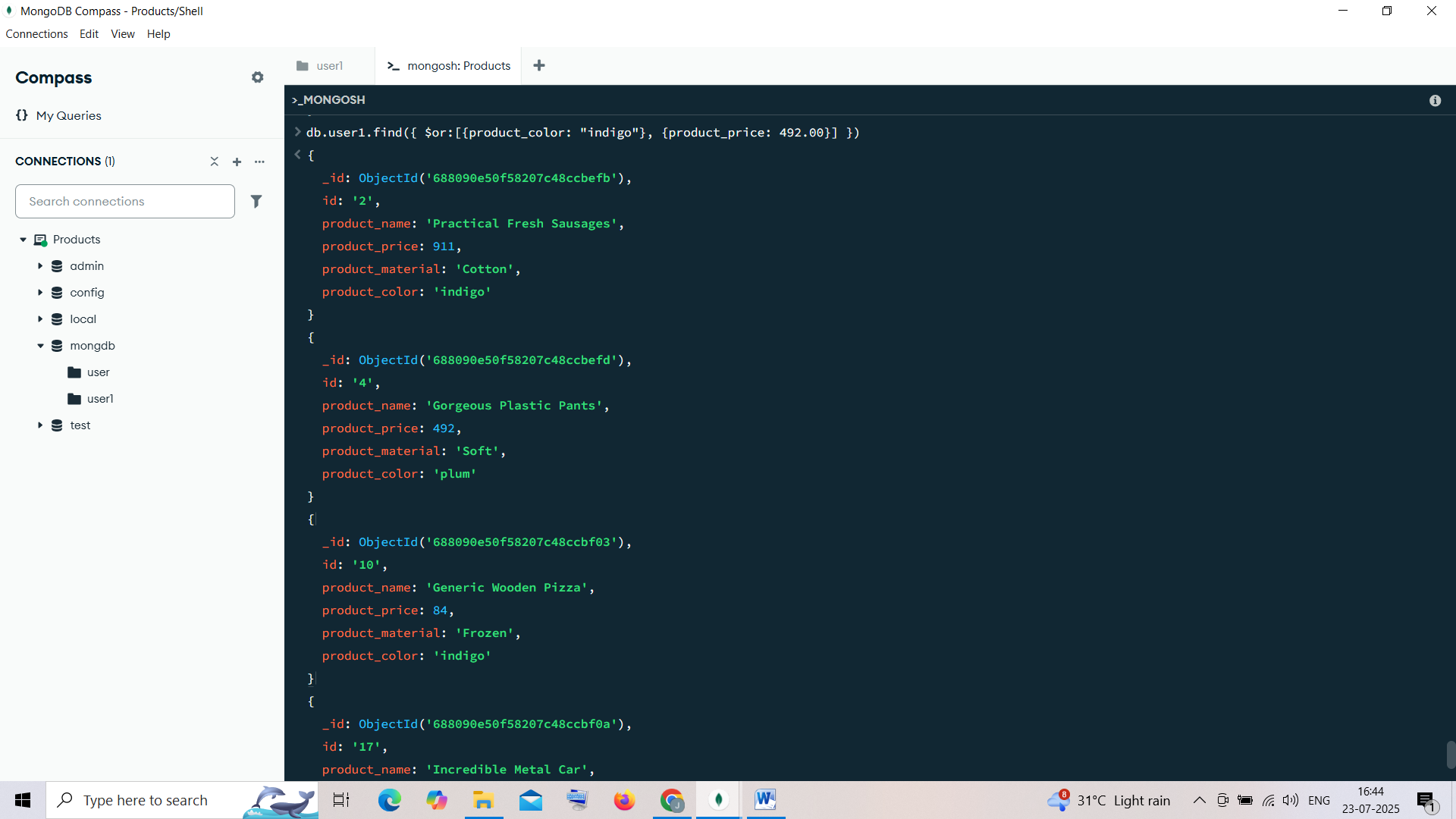
### 8. ****Find all products which contain the value**** soft ****in**** product\_material

db.user1.find({ product\_material: "Soft”})



### 9. ****Find products which contain product color**** indigo ****and product price**** 492.00

db.user1.find({ $or:[{product\_color: "indigo"}, {product\_price: 492.00}] })



### 10. ****Delete the products which have duplicate**** price ****values****

This requires aggregation to find duplicates, then delete them.

#### Step 1: Find duplicate prices

db.user1.aggregate([

{ $group: {

\_id: "$product\_price",

count: { $sum: 1 },

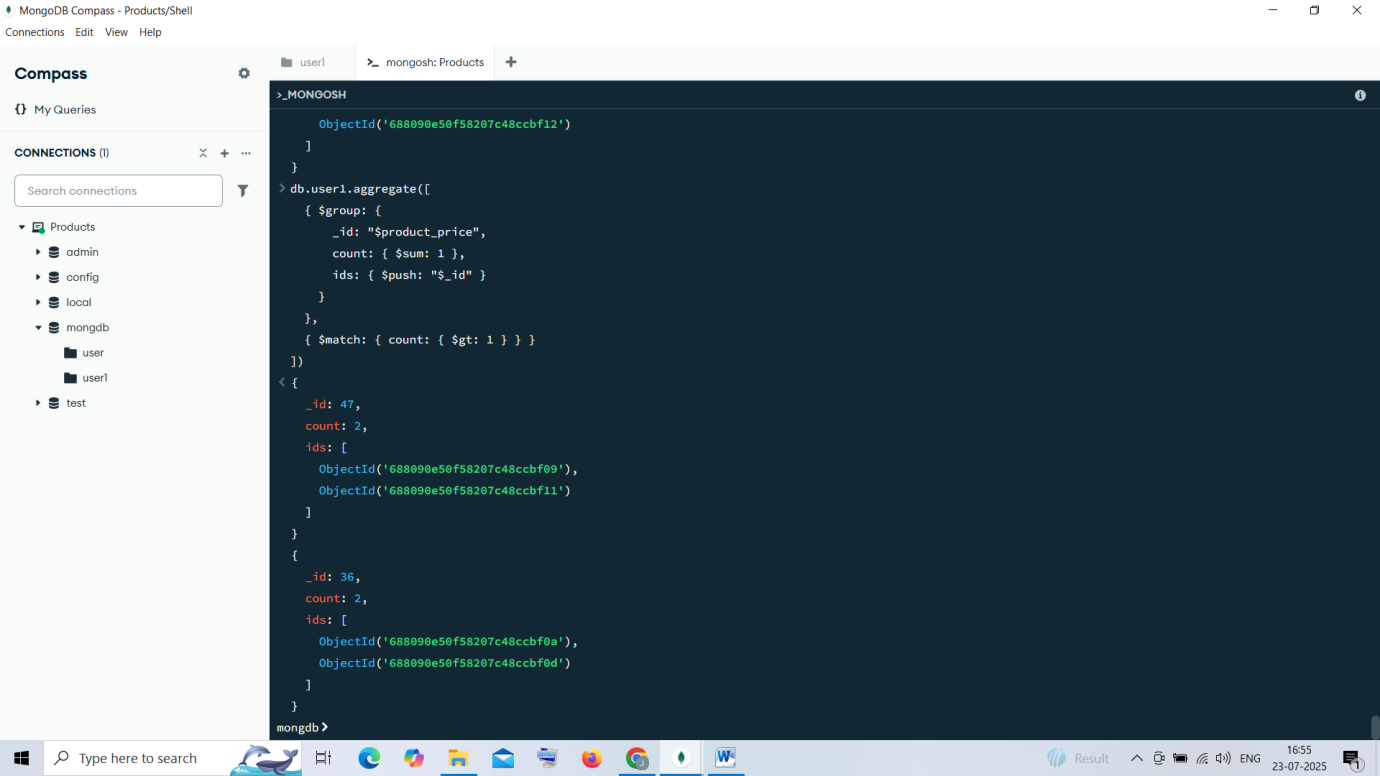
ids: { $push: "$\_id" }

}

},

{ $match: { count: { $gt: 1 } } }

])



#### Step 2: Delete duplicates

db.user1.aggregate([

{ $group: {

\_id: "$product\_price",

ids: { $addToSet: "$\_id" },

count: { $sum: 1 }

}

},

{ $match: { count: { $gt: 1 } } }

]).forEach(doc => {

doc.ids.shift(); // keep one

db.user1.deleteMany({ \_id: { $in: doc.ids } });

});

