**MongoDB Task-1**

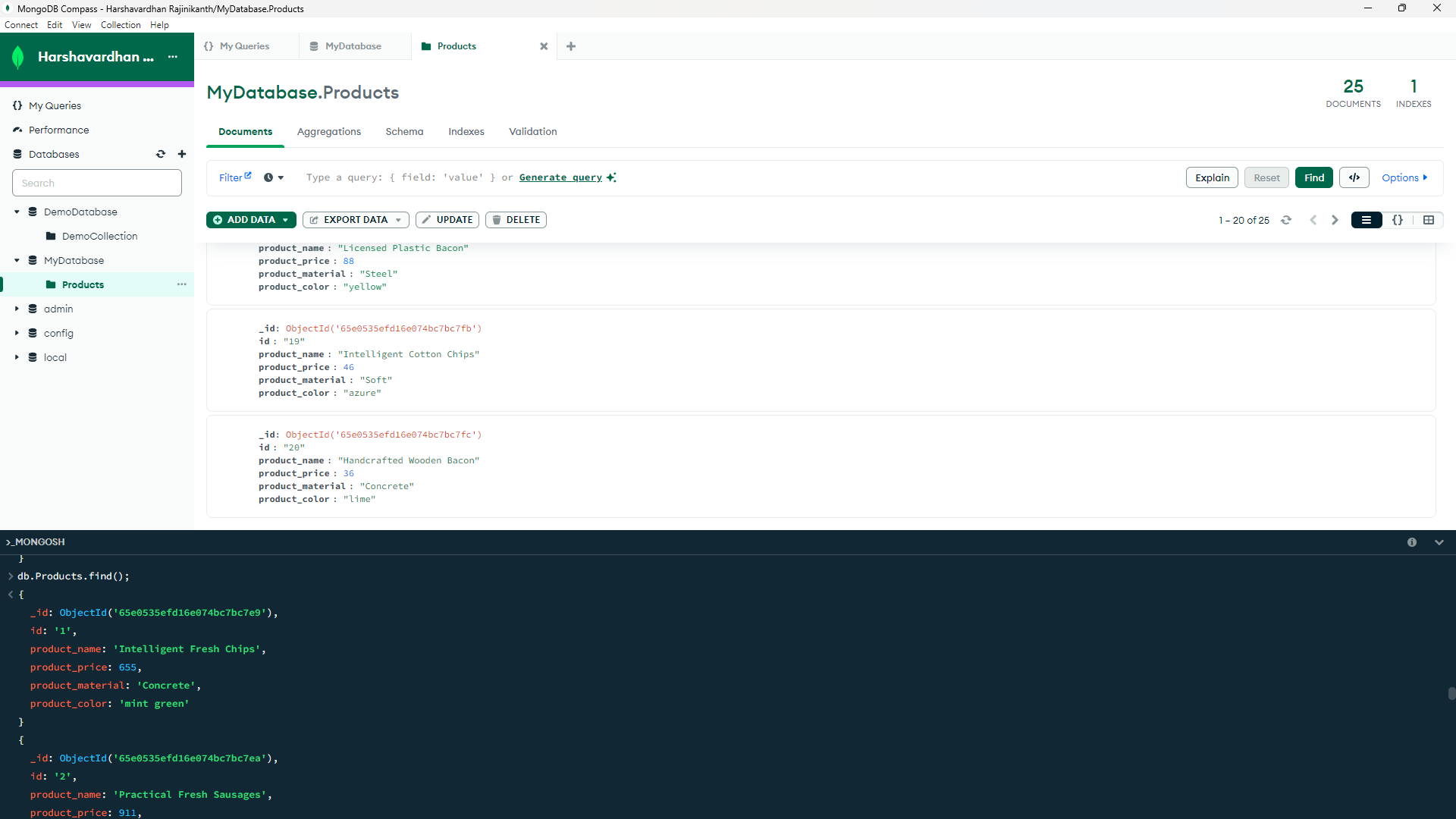
**Database Name: MyDatabase**

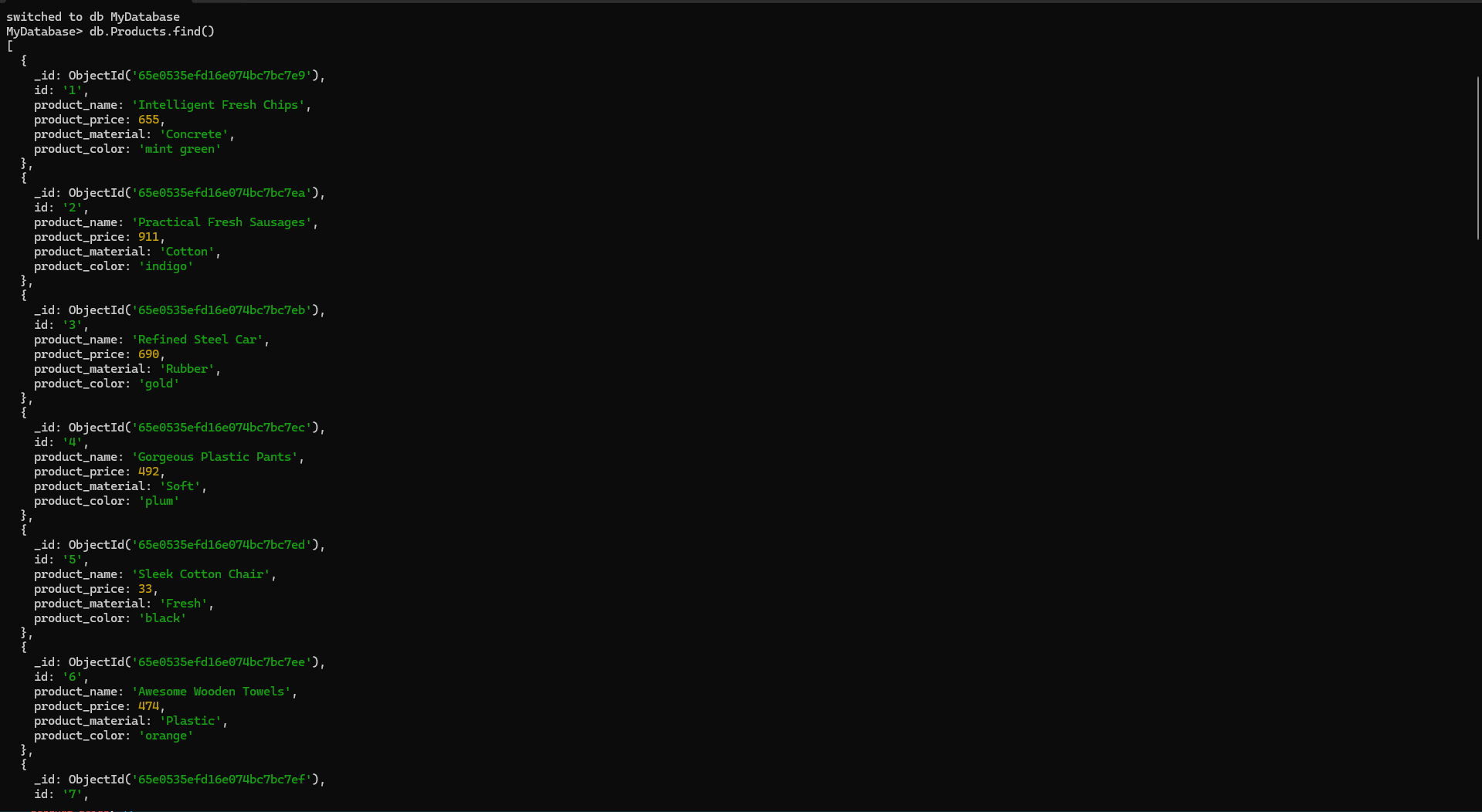
**Collection Name: Products**

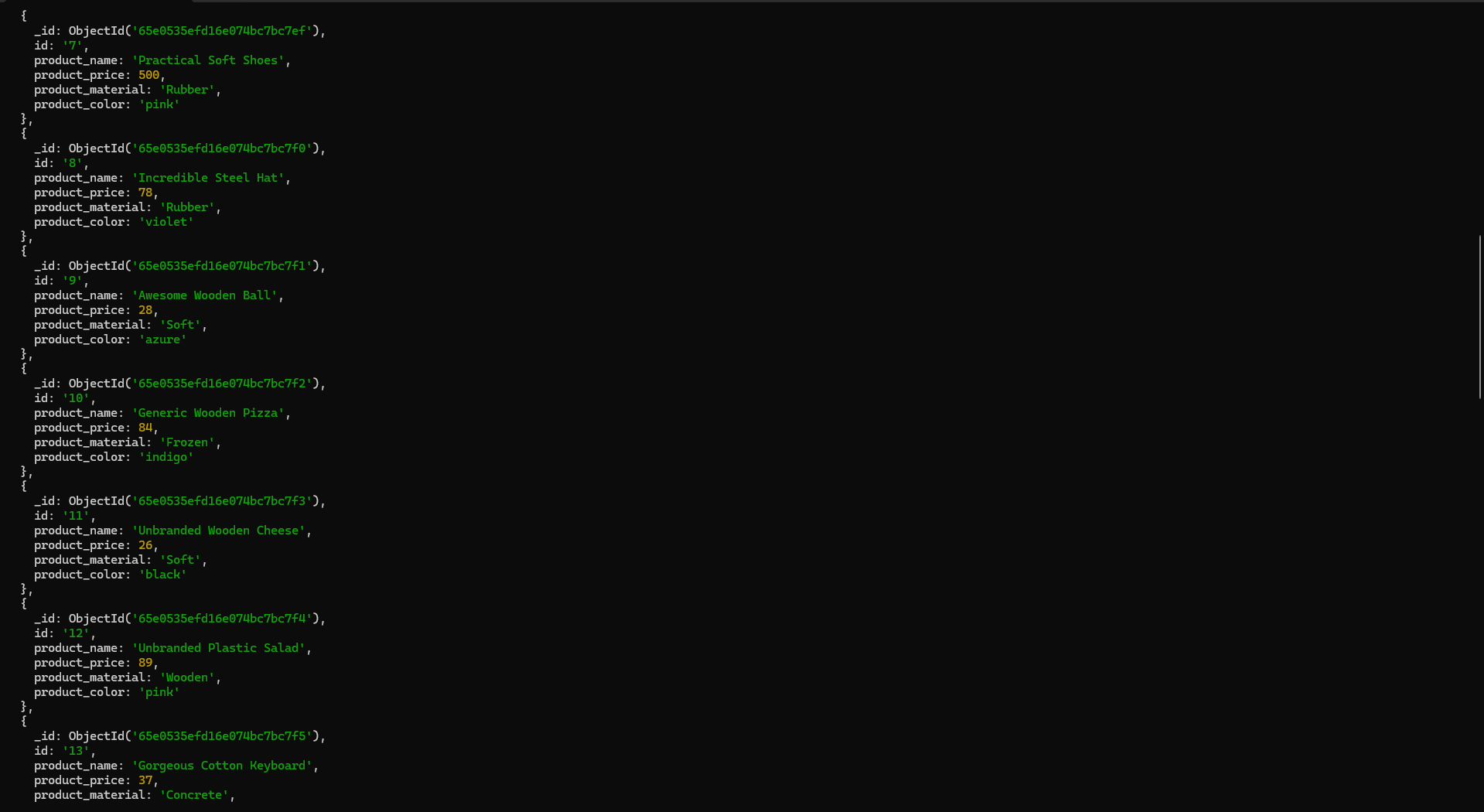
1. **Find all the information about each products.**

**Query: db.Products.find();**

**Output:**

****





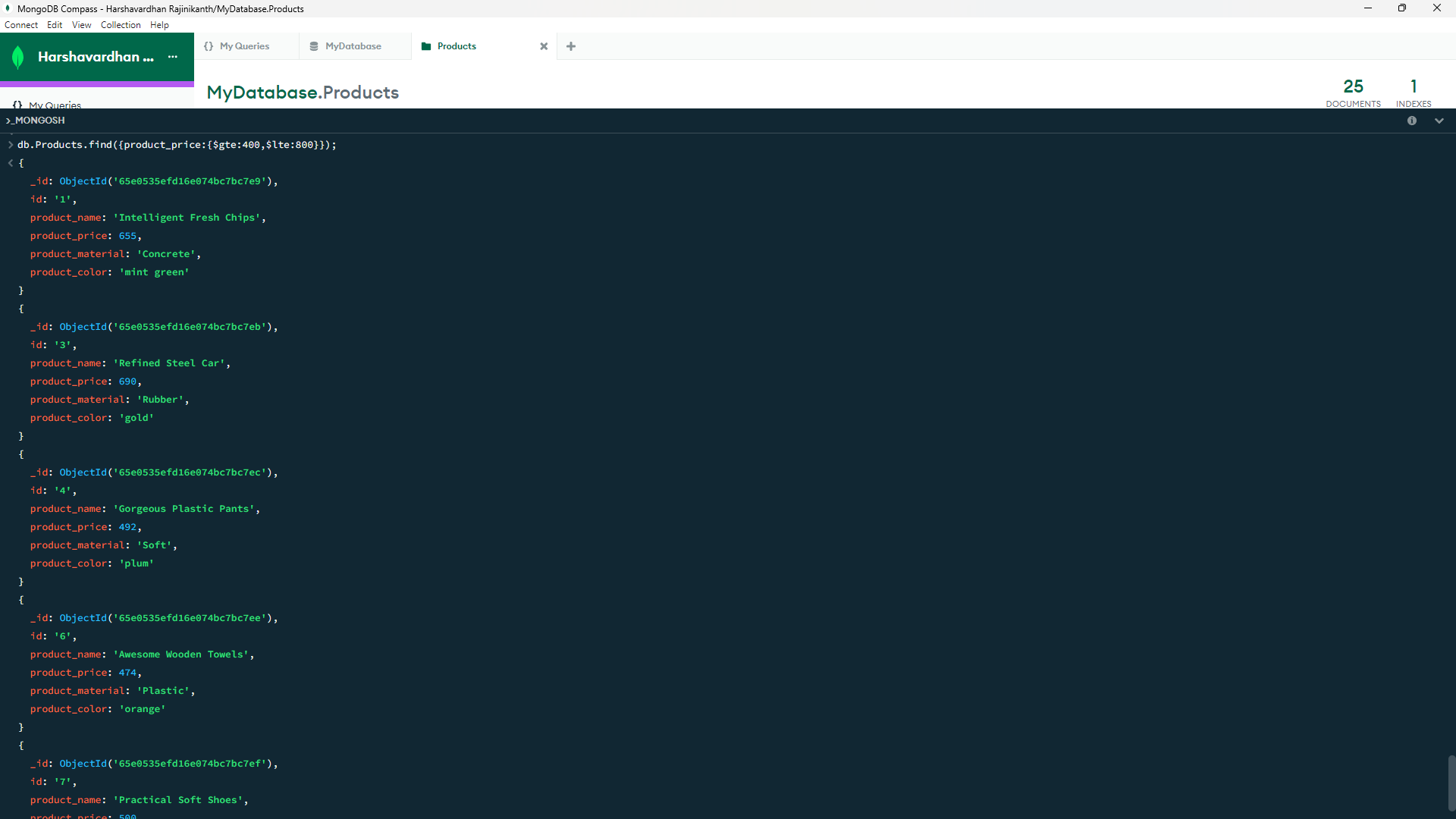




1. **Find the product price which are between 400 to 800**

**Query: db.Products.find({‘product\_price’:{$gte:400,$lte:800}});**

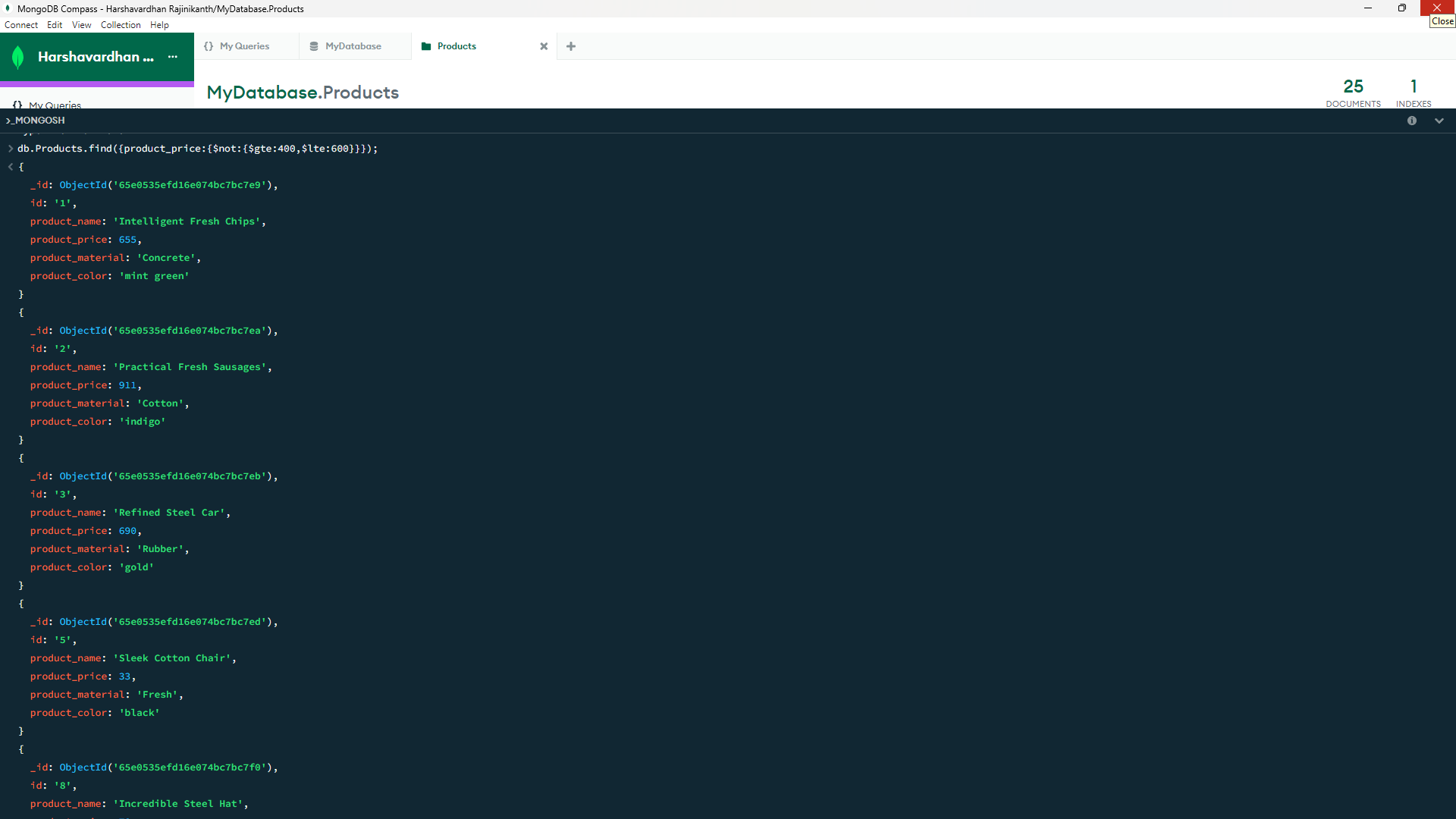
**Output:**

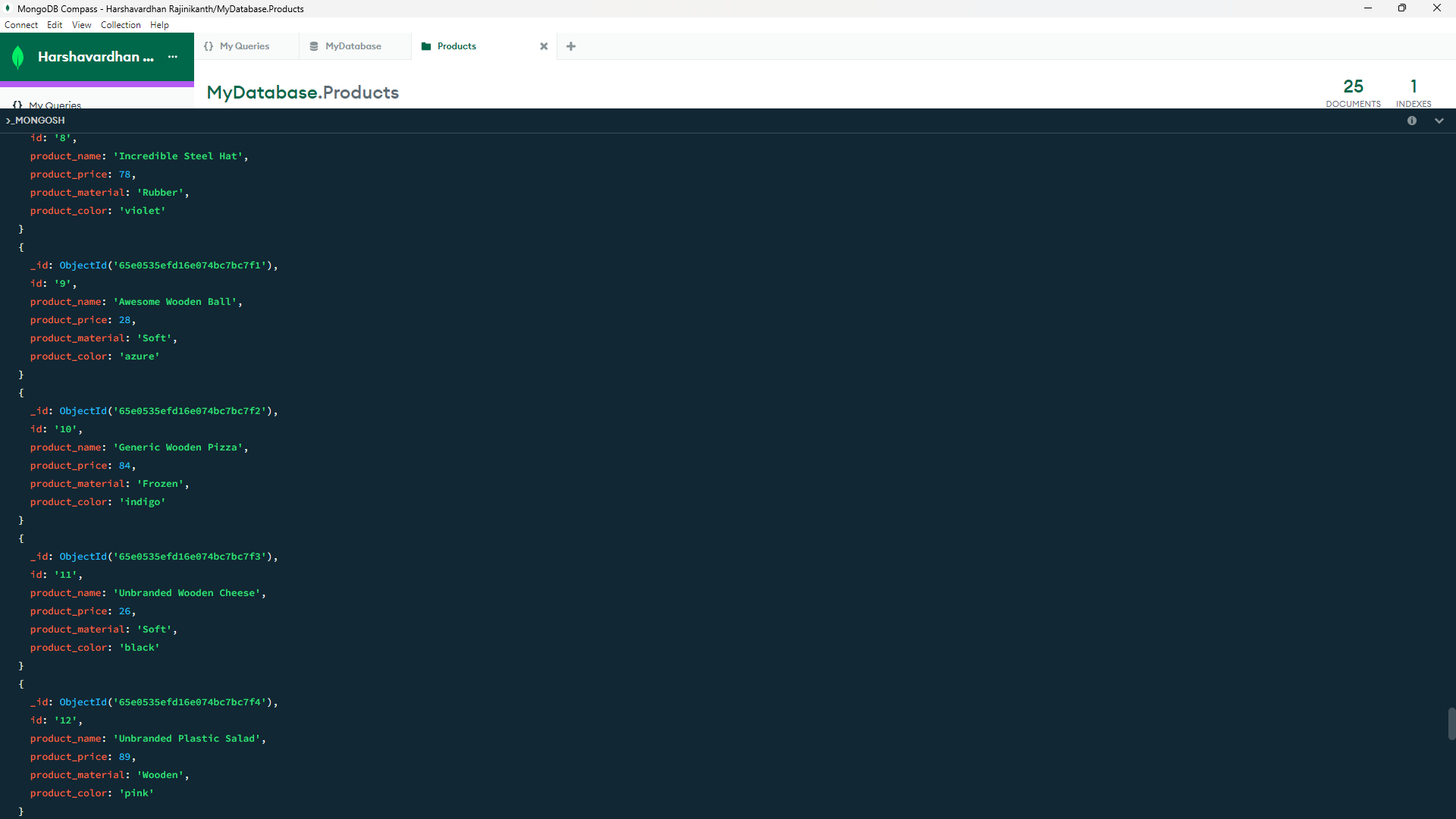
****

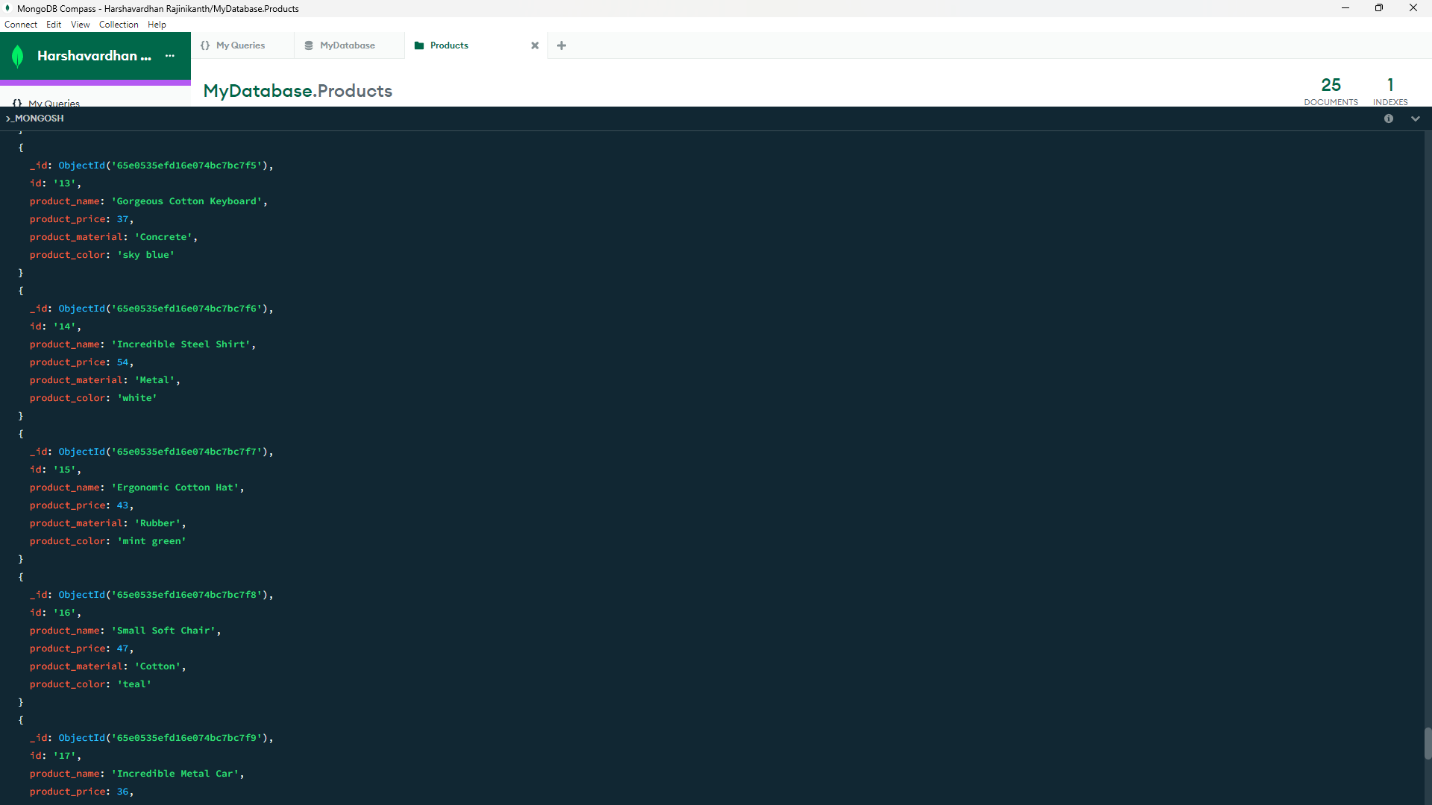
1. **Find the product price which are not between 400 to 600**

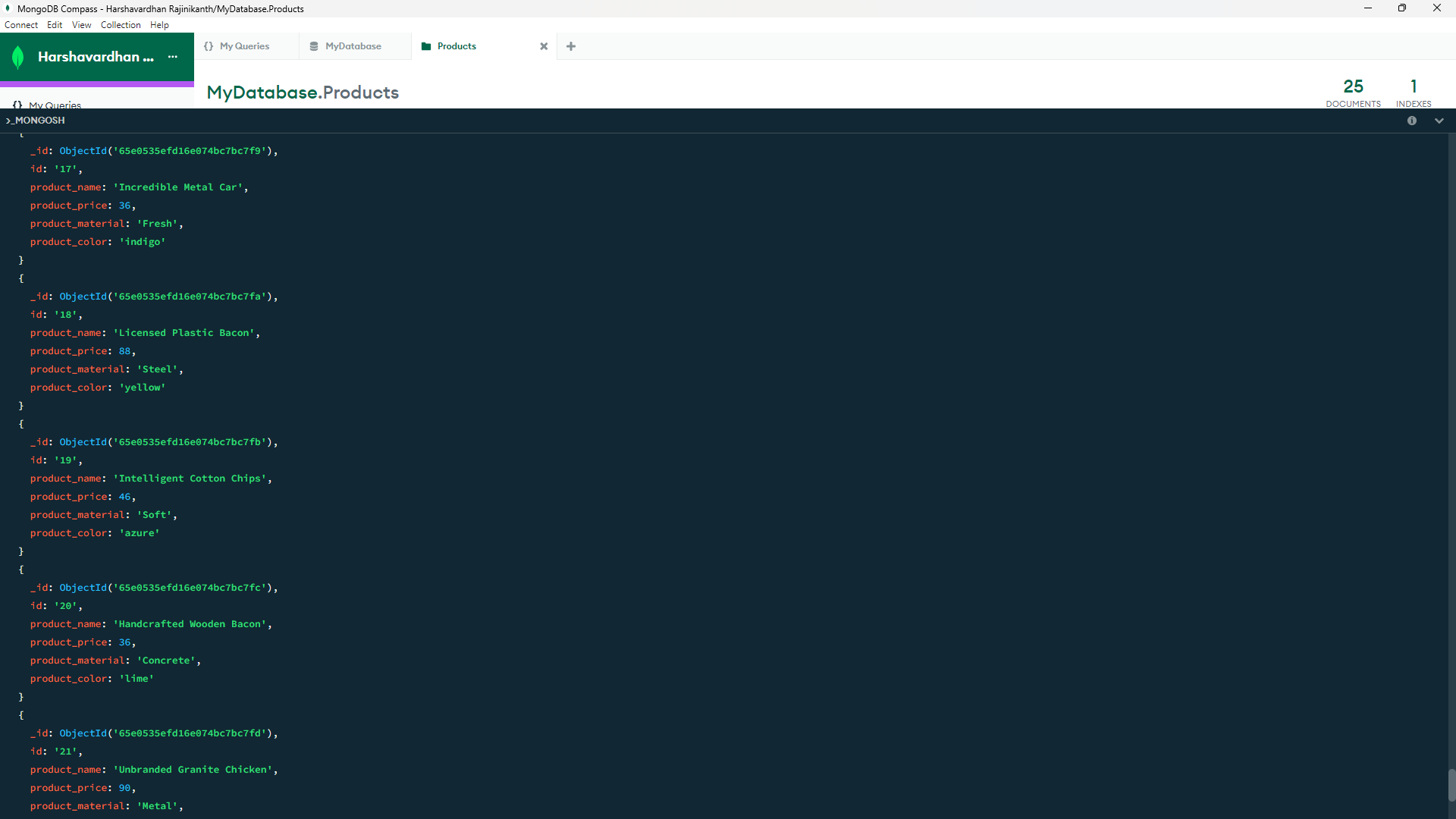
**Query:db.Products.find({‘product\_price’:{$not**:**{$gte:400,$lte:600}}});**

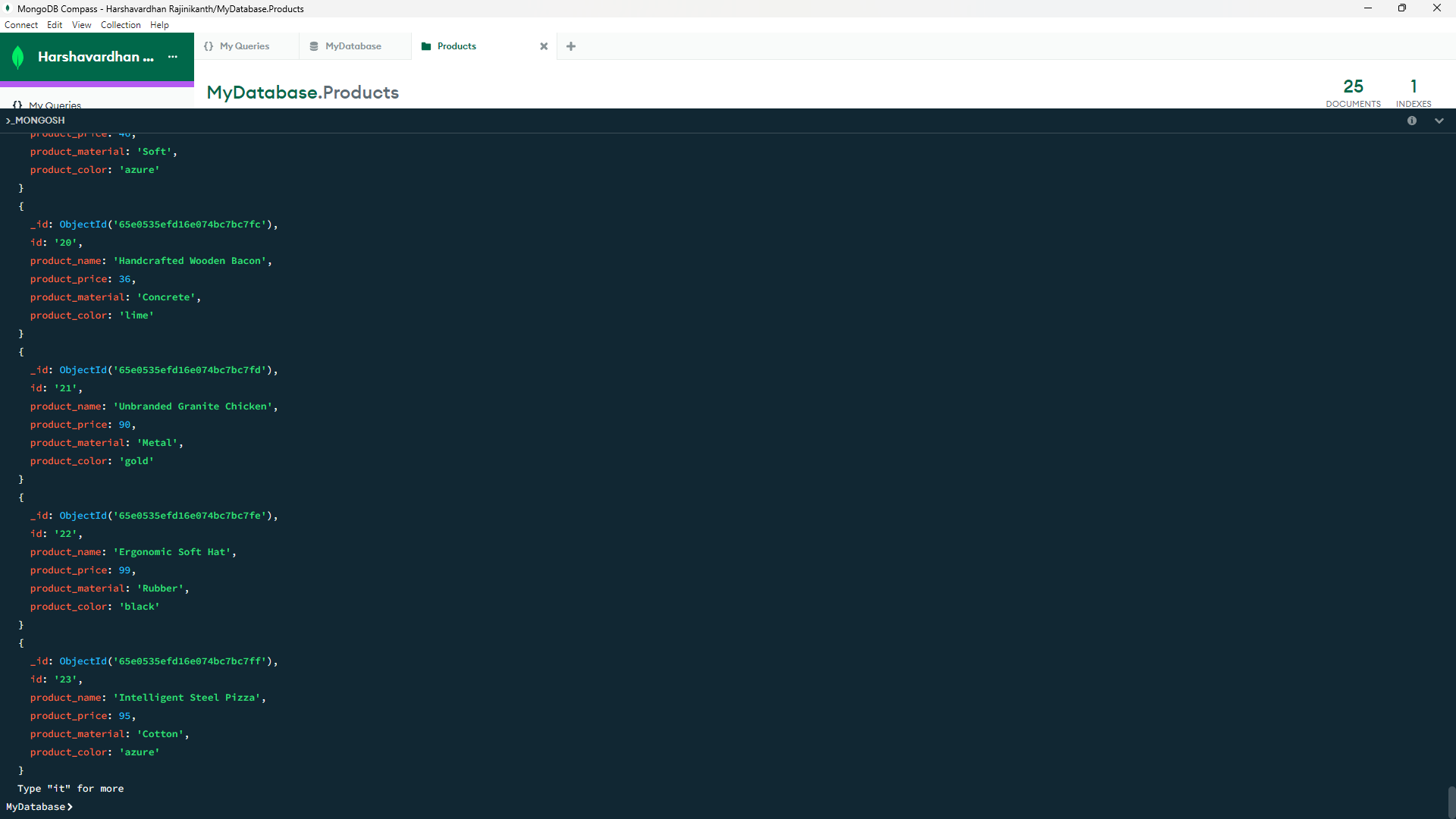
**Output:**

****

****

****

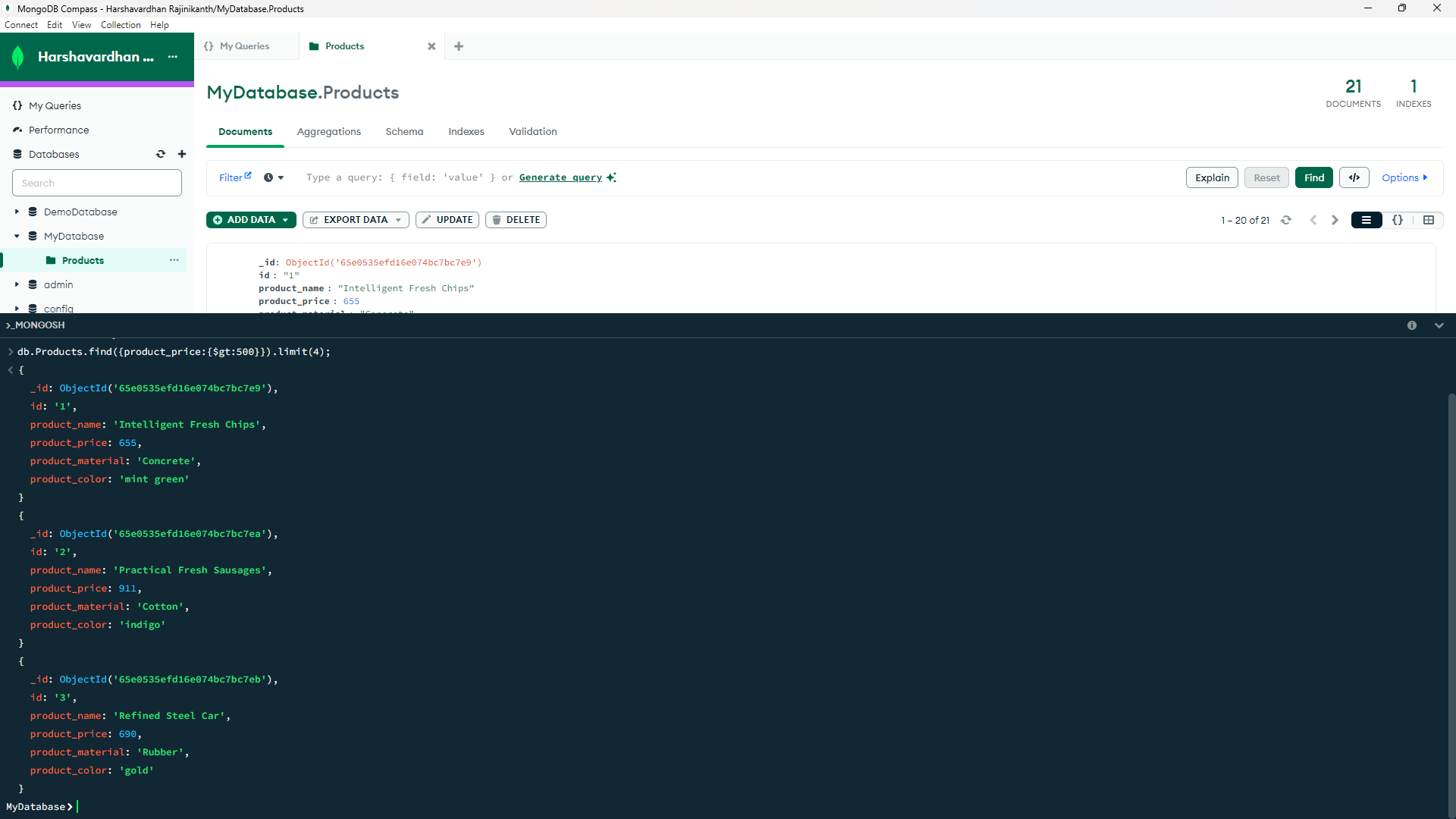
****

****

1. **List the four product which are greater than 500 in price**

**Query:** **db.Products.find({‘product\_price’:{$gt:500}})** **.limit(4);**

**Output:**

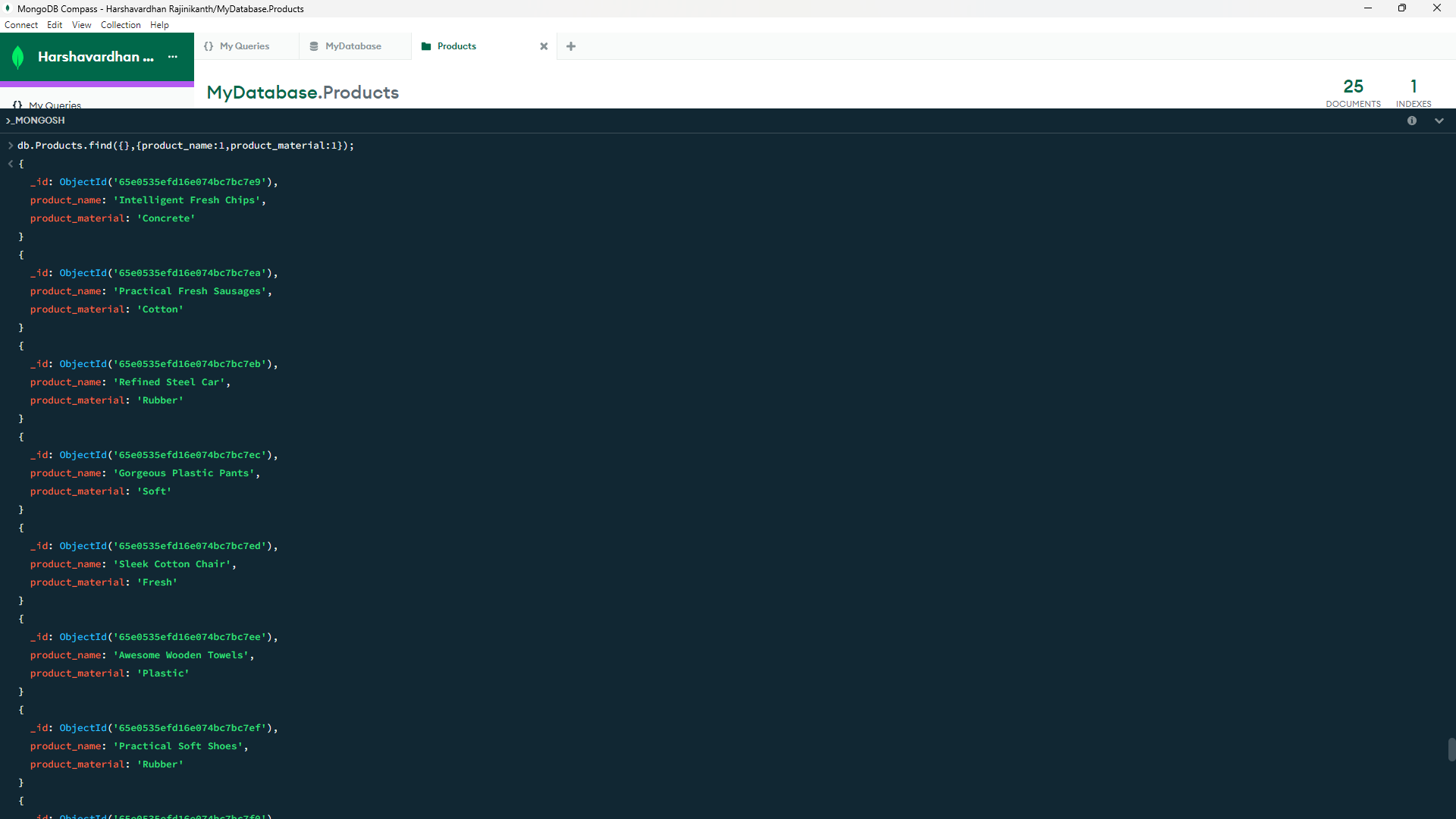
****

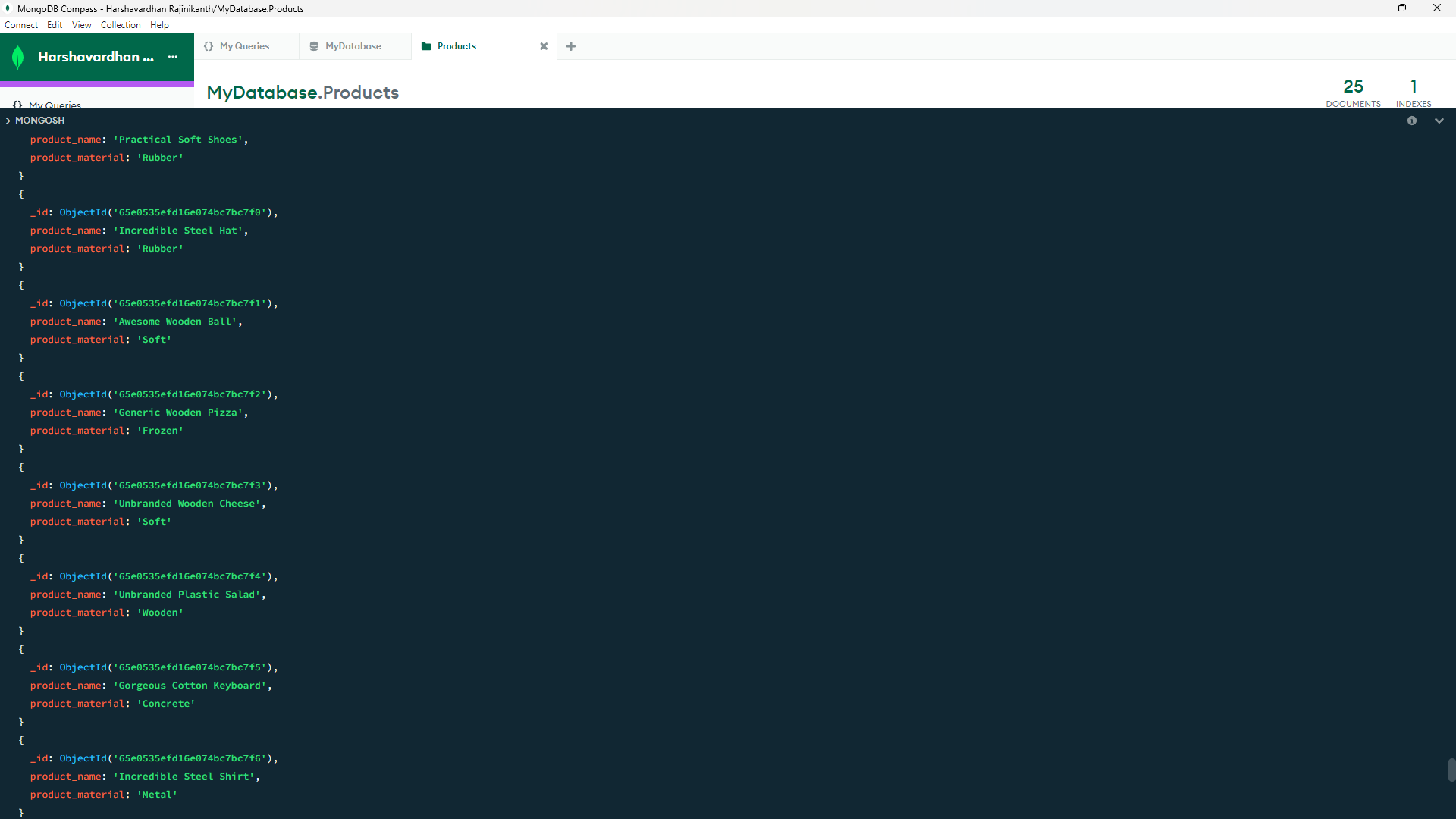
1. **Find the product name and product material of each products**

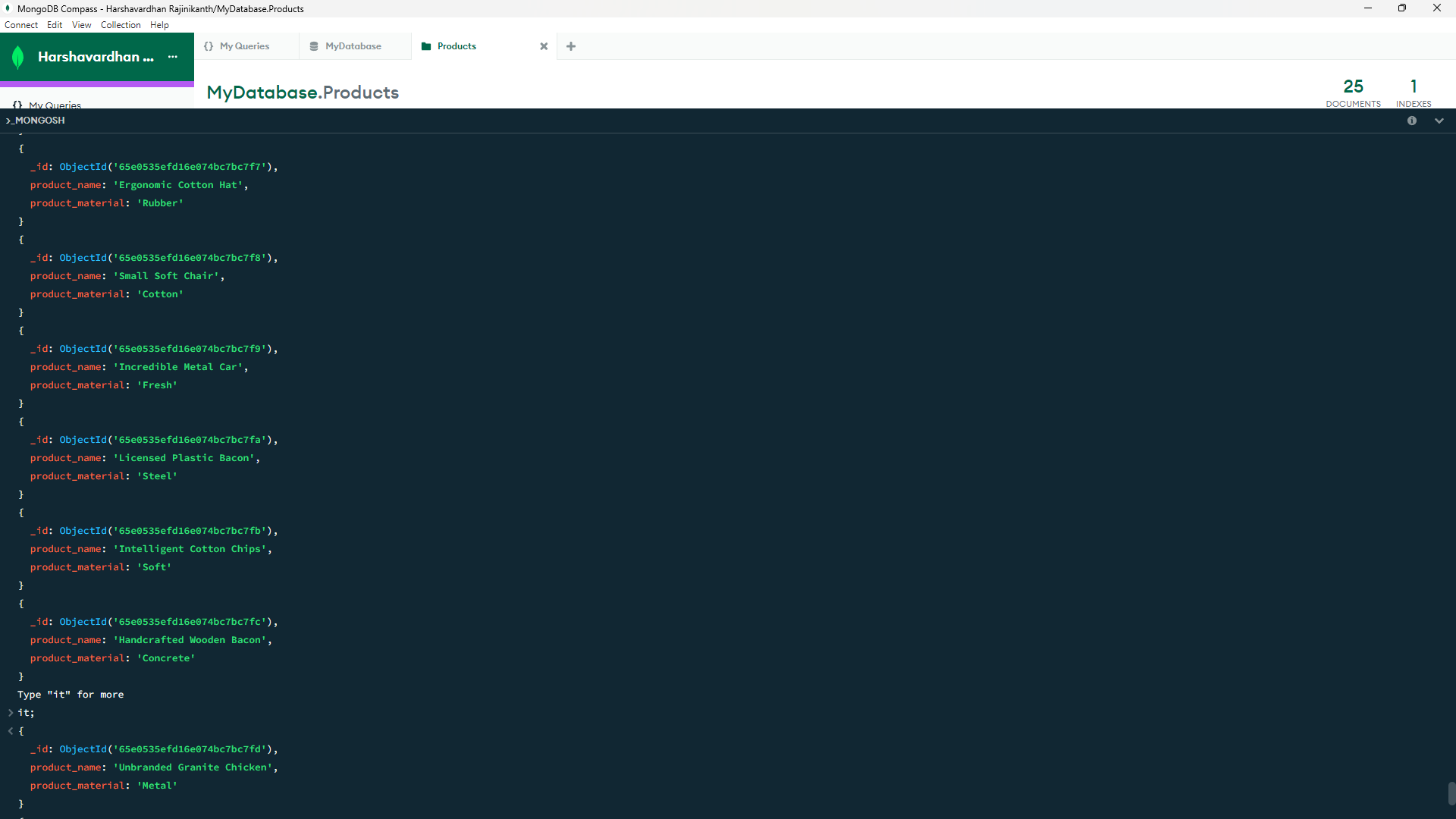
**Query:**

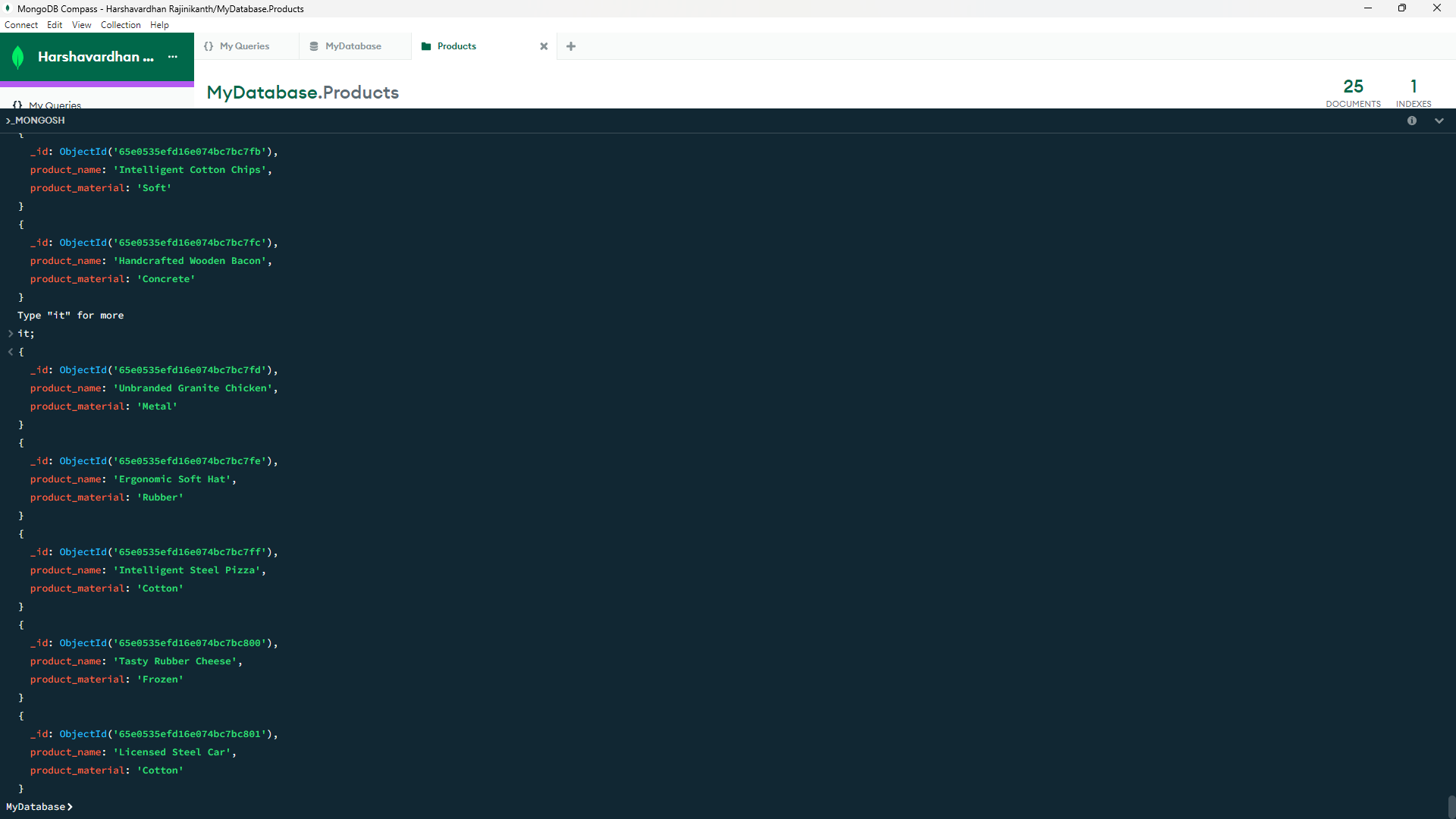
**db.Products.find({},{‘product\_name’:1,’product\_material’:1});**

**Output:**

****

****

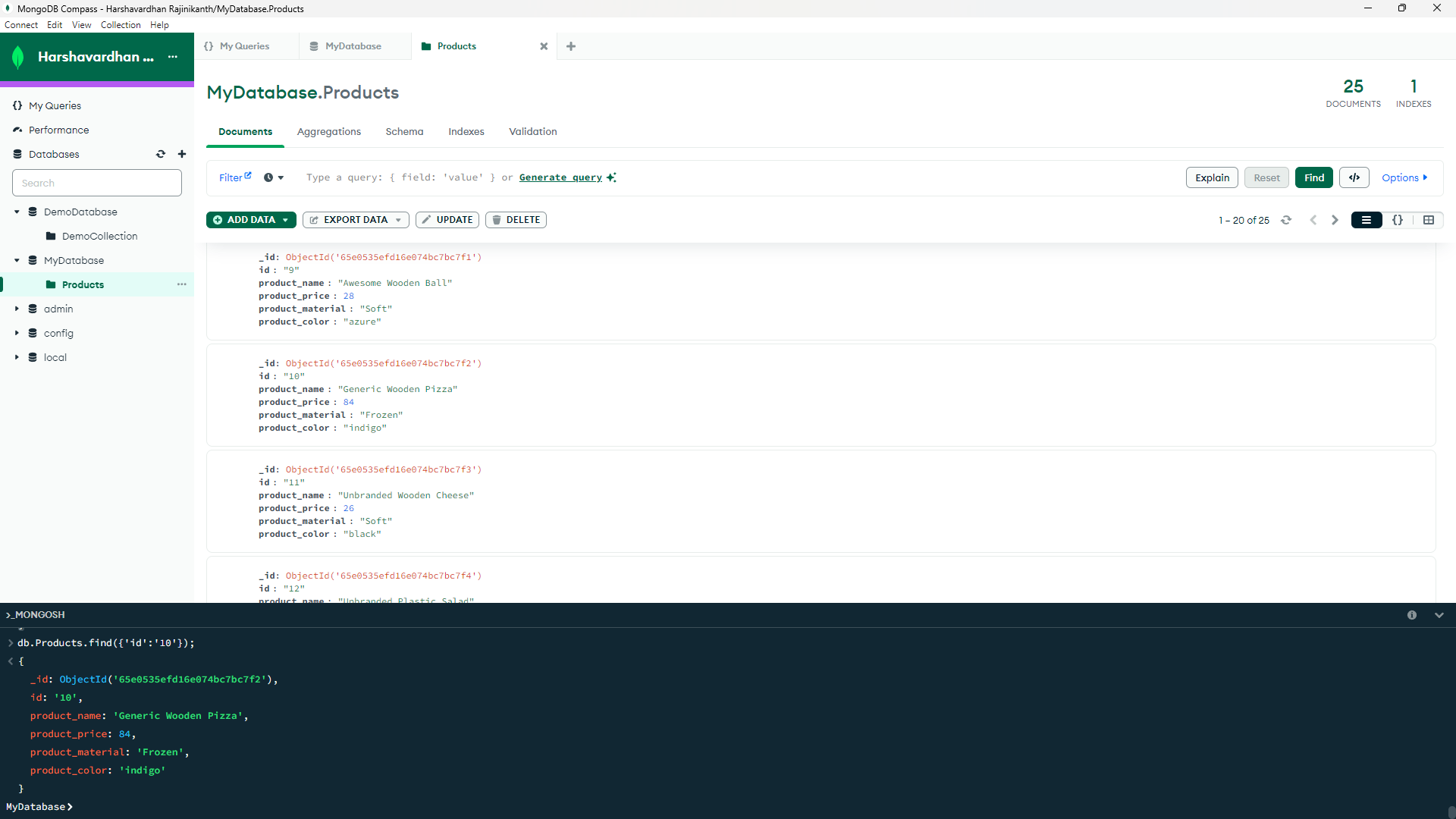
****

****

1. **Find the product with a row id of 10**

**Query:** **db.Products.find({'id':'10'});**

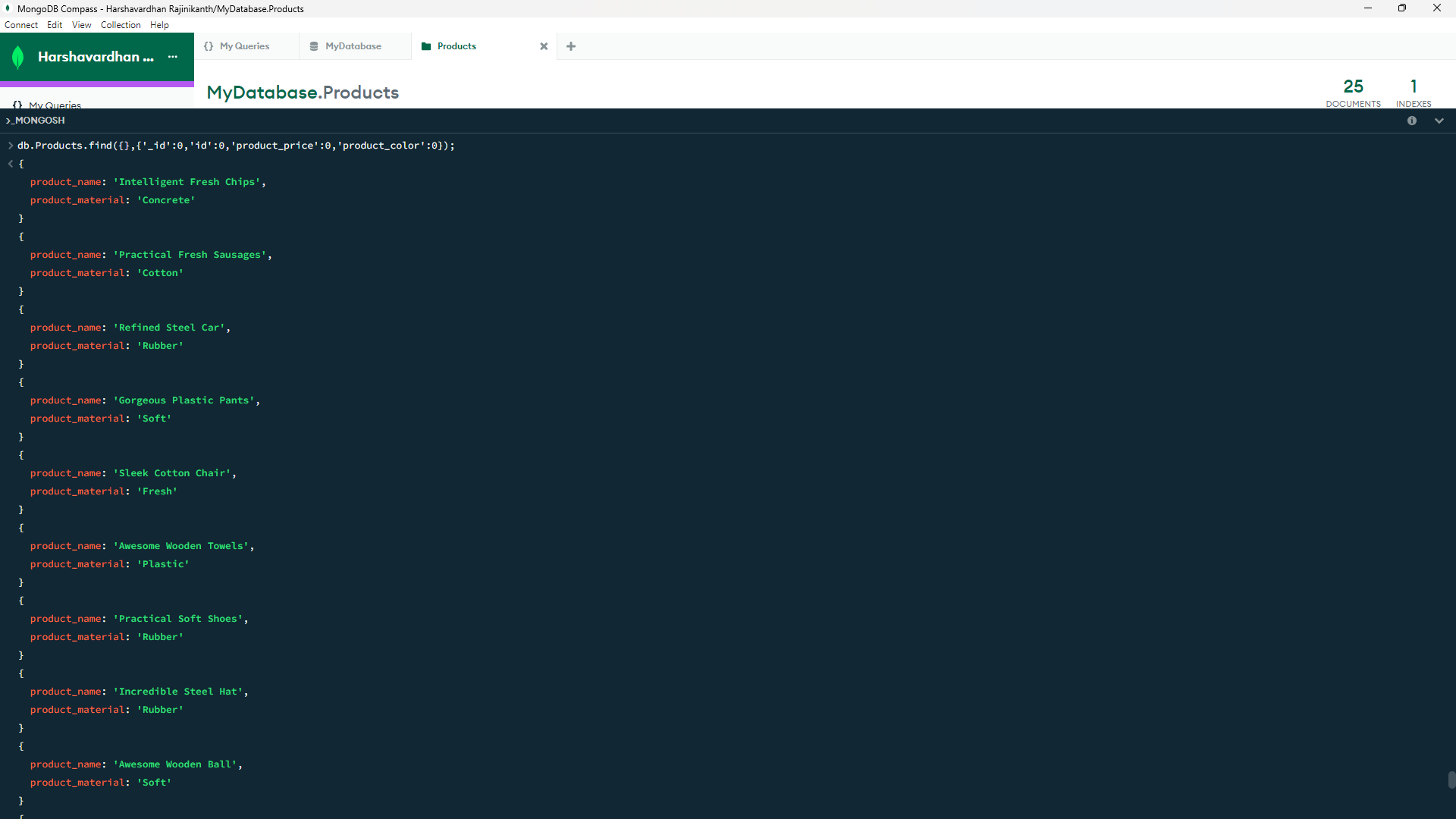
**Output:**

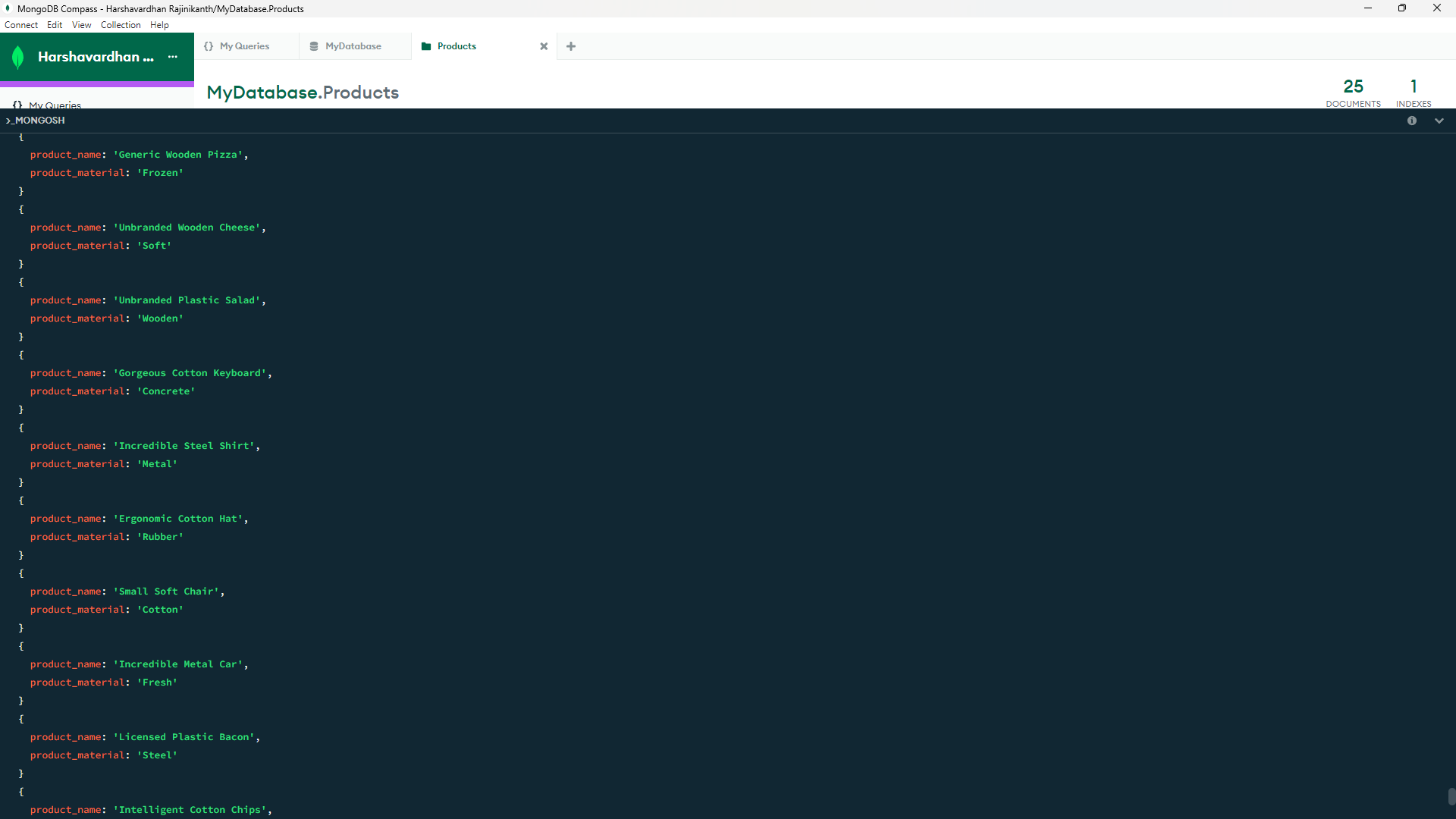
****

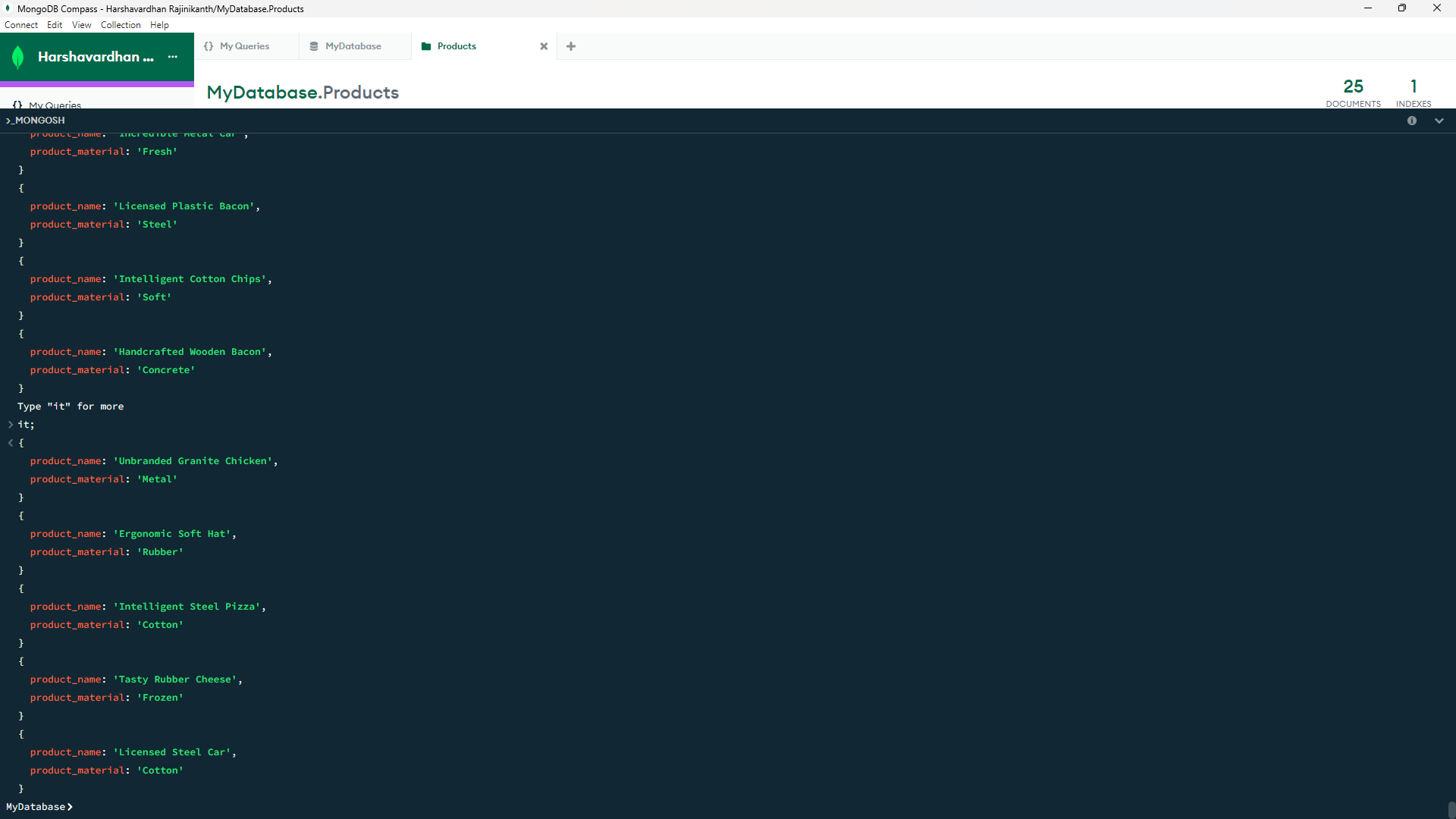
1. **Find only the product name and product material**

**Query:db.Products.find({},{'\_id':0,'id':0,'product\_price':0,'product\_color':0});**

**Output:**

****

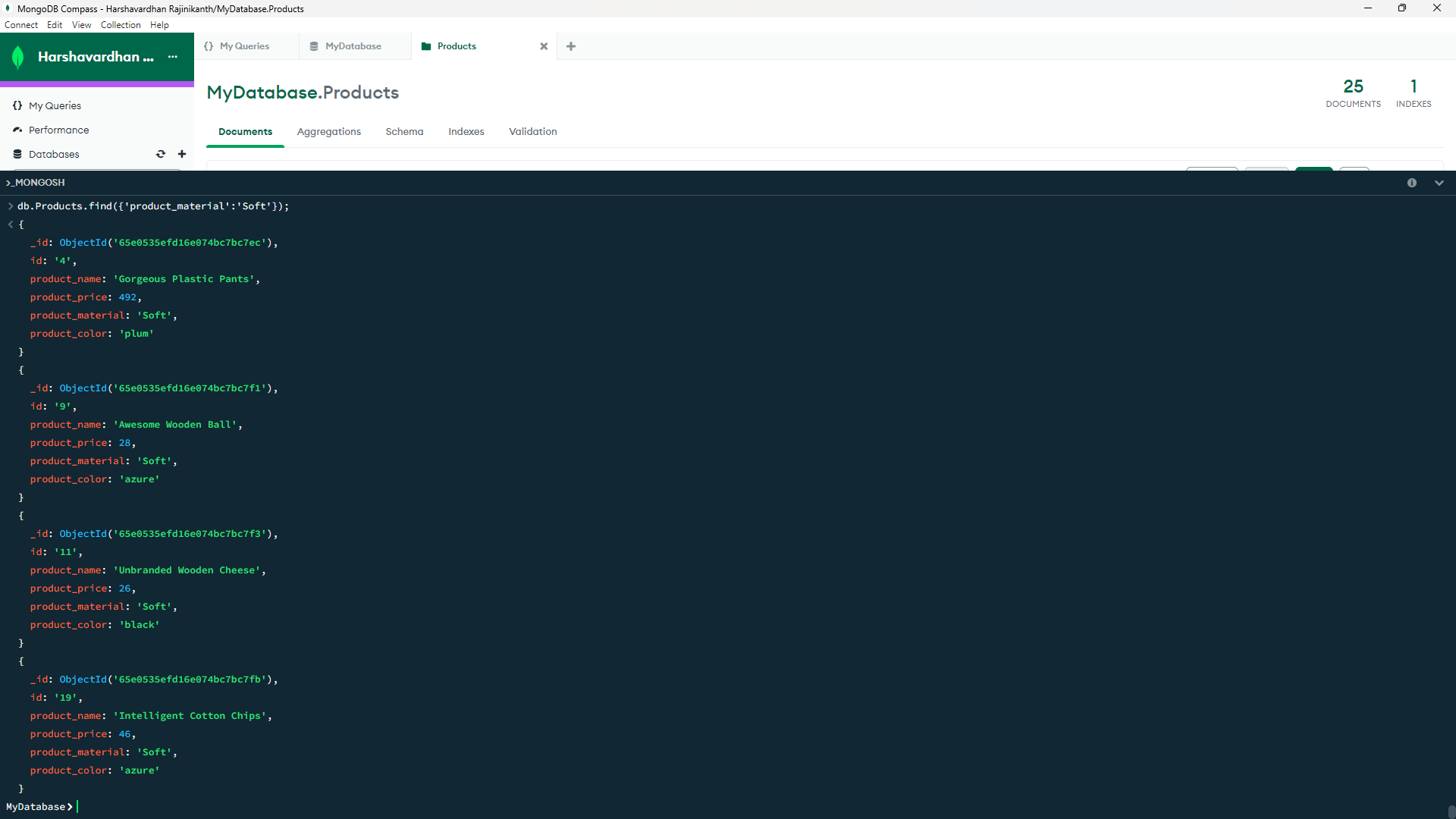
****

****

1. **Find all products which contain the value of soft in product material**

**Query:** **db.Products.find({'product\_material':'Soft'});**

**Output:**

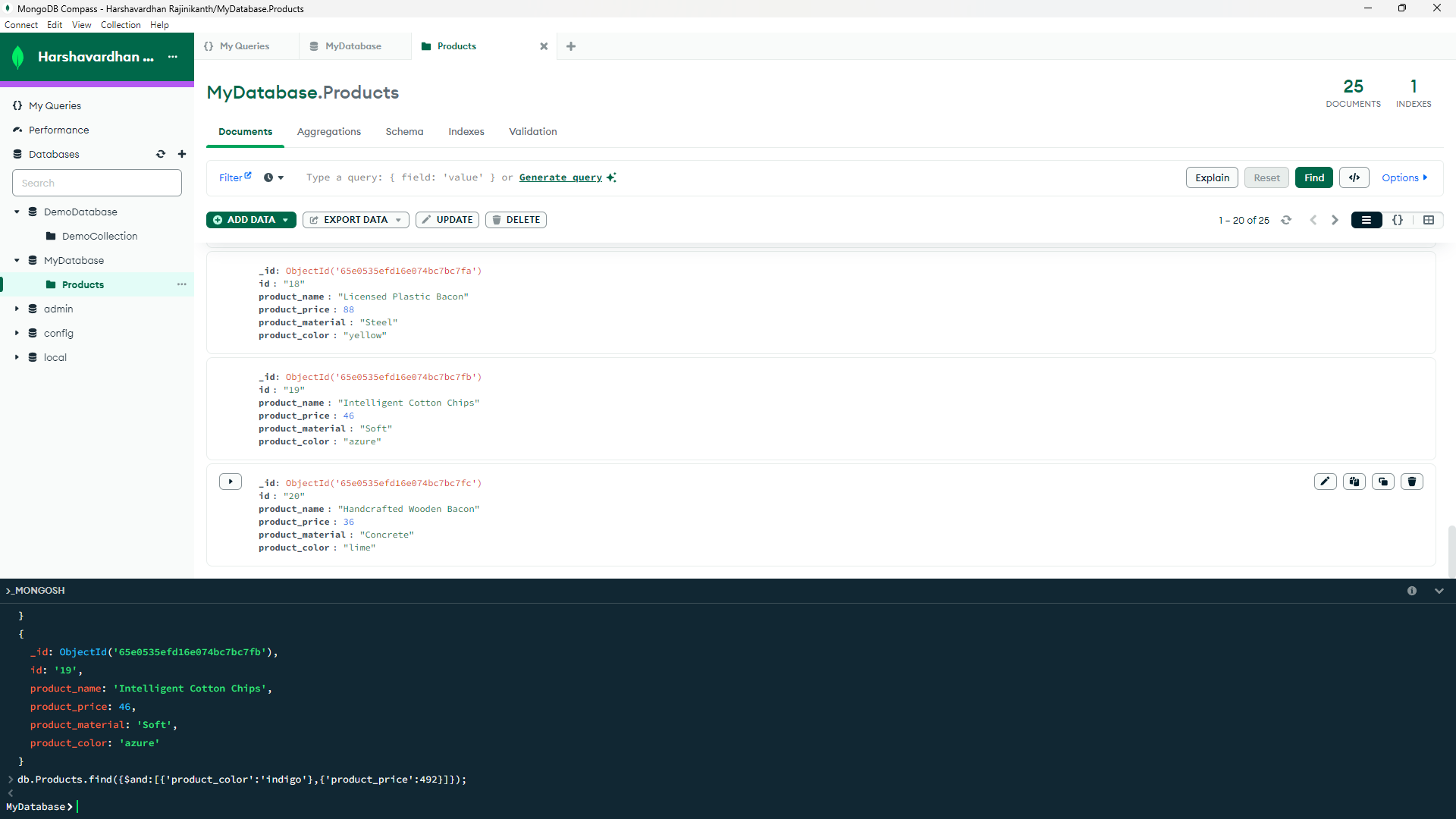
****

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

**Query:**

**db.Products.find({$and:[{'product\_color':'indigo'},{'product\_price':492}]});**

**Output: No products present according to the condition.**

****

1. **Delete the products which product price value are same**

**Query:**

db.Products.aggregate([

{

$group: {

\_id: '$product\_price',

count: { $sum: 1 }

}

},

{

$match: {

count: { $gt: 1 }

}

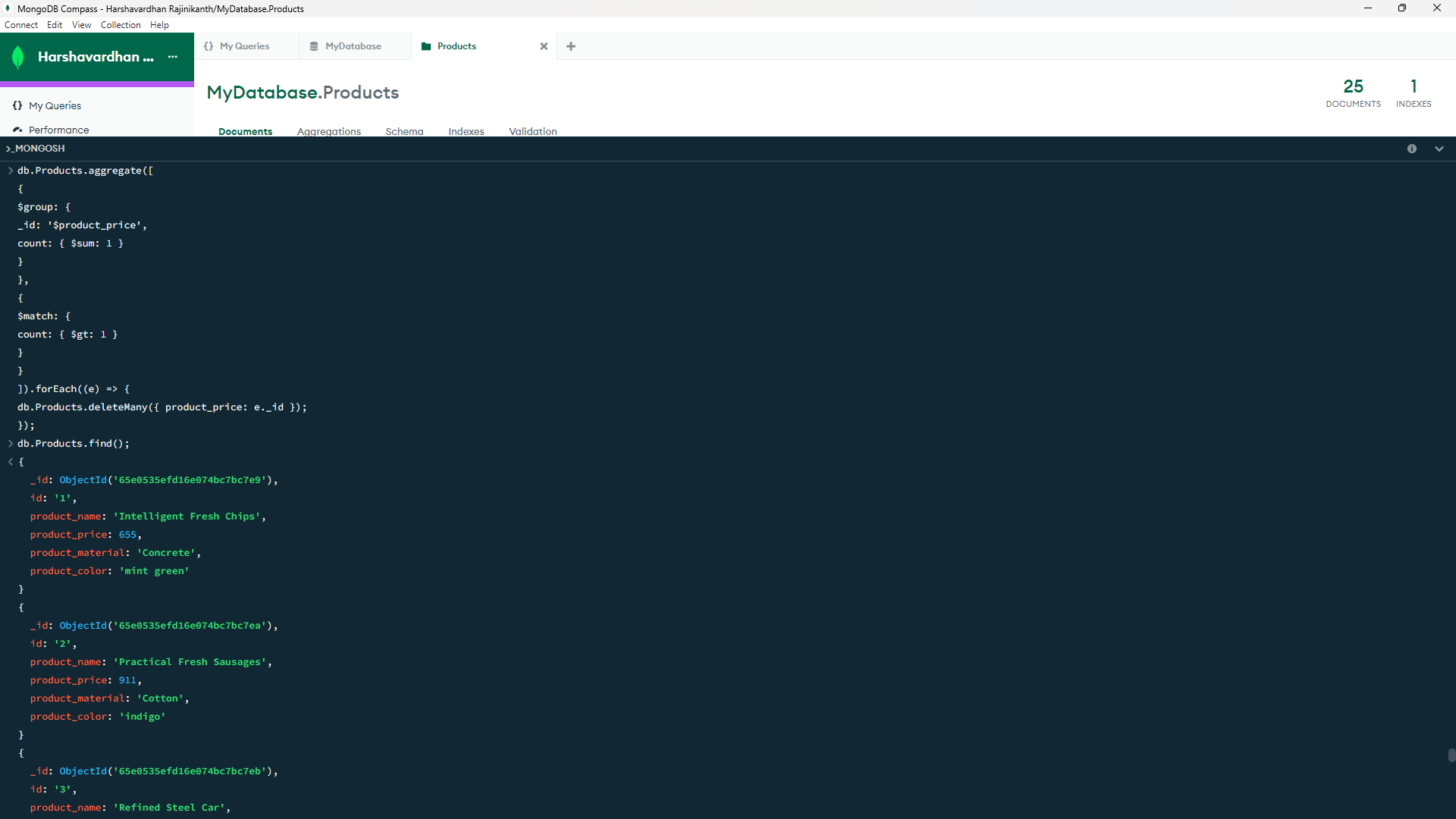
}

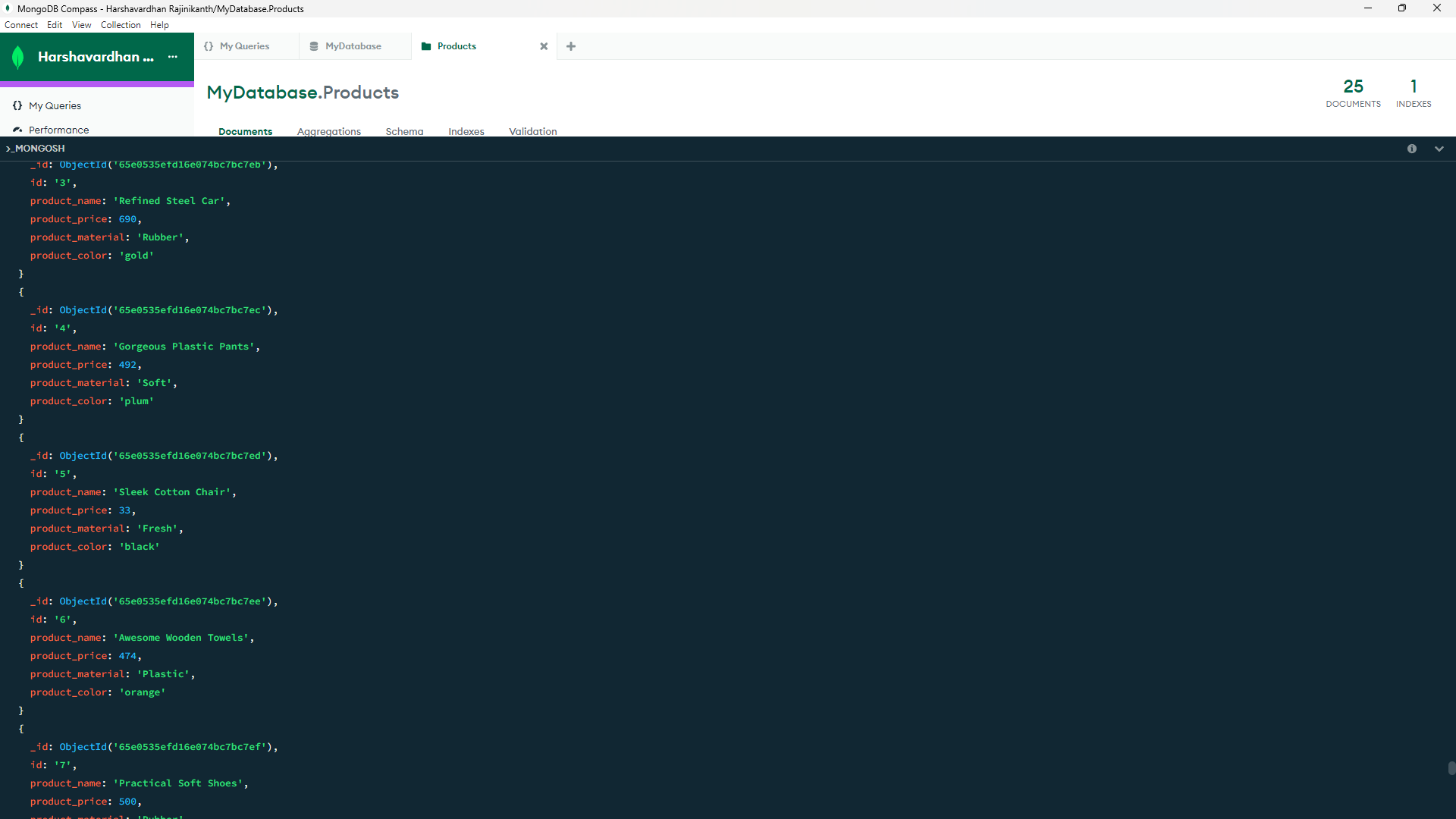
]).forEach((e) => {

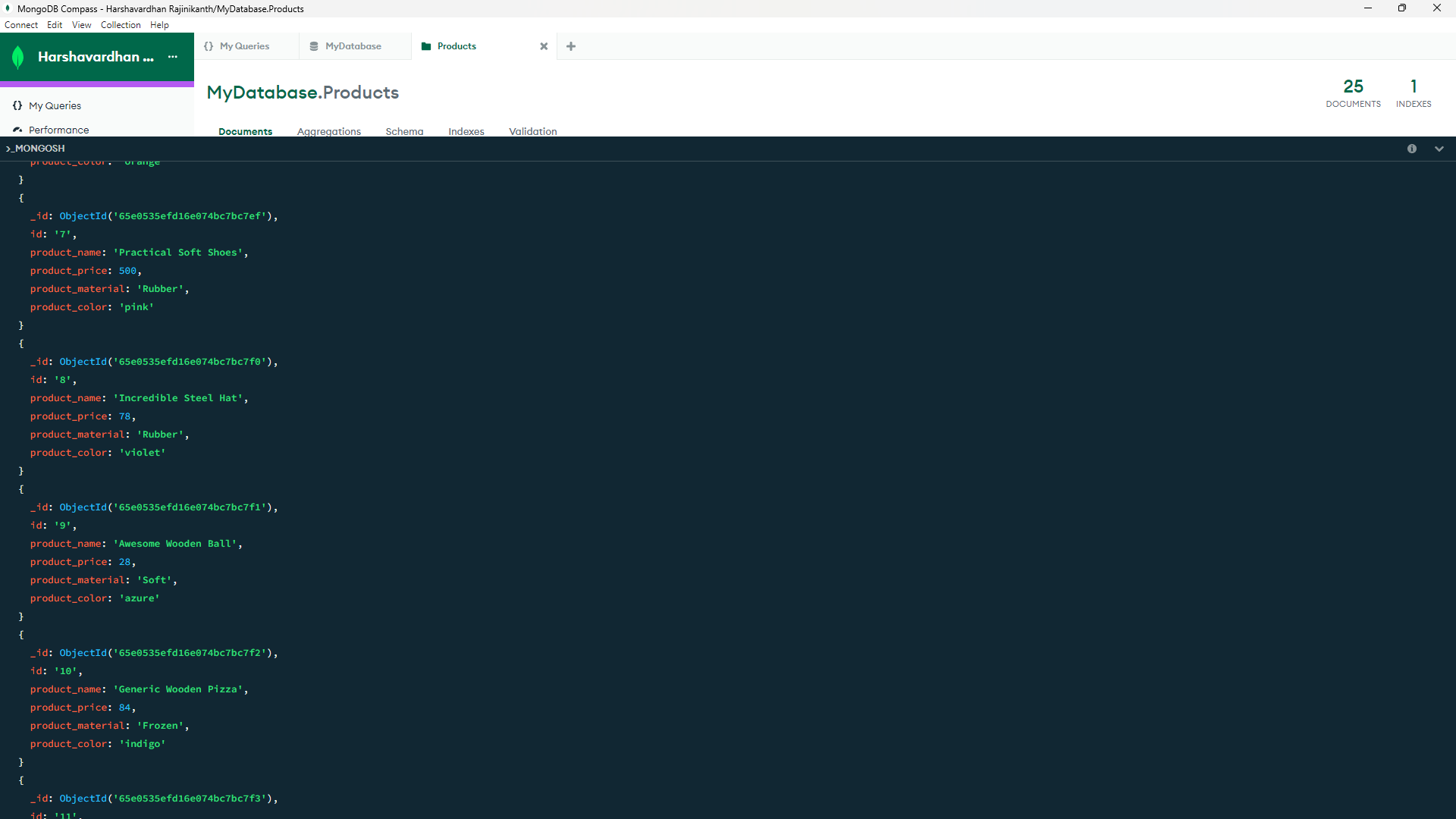
db.Products.deleteMany({ product\_price: e.\_id });

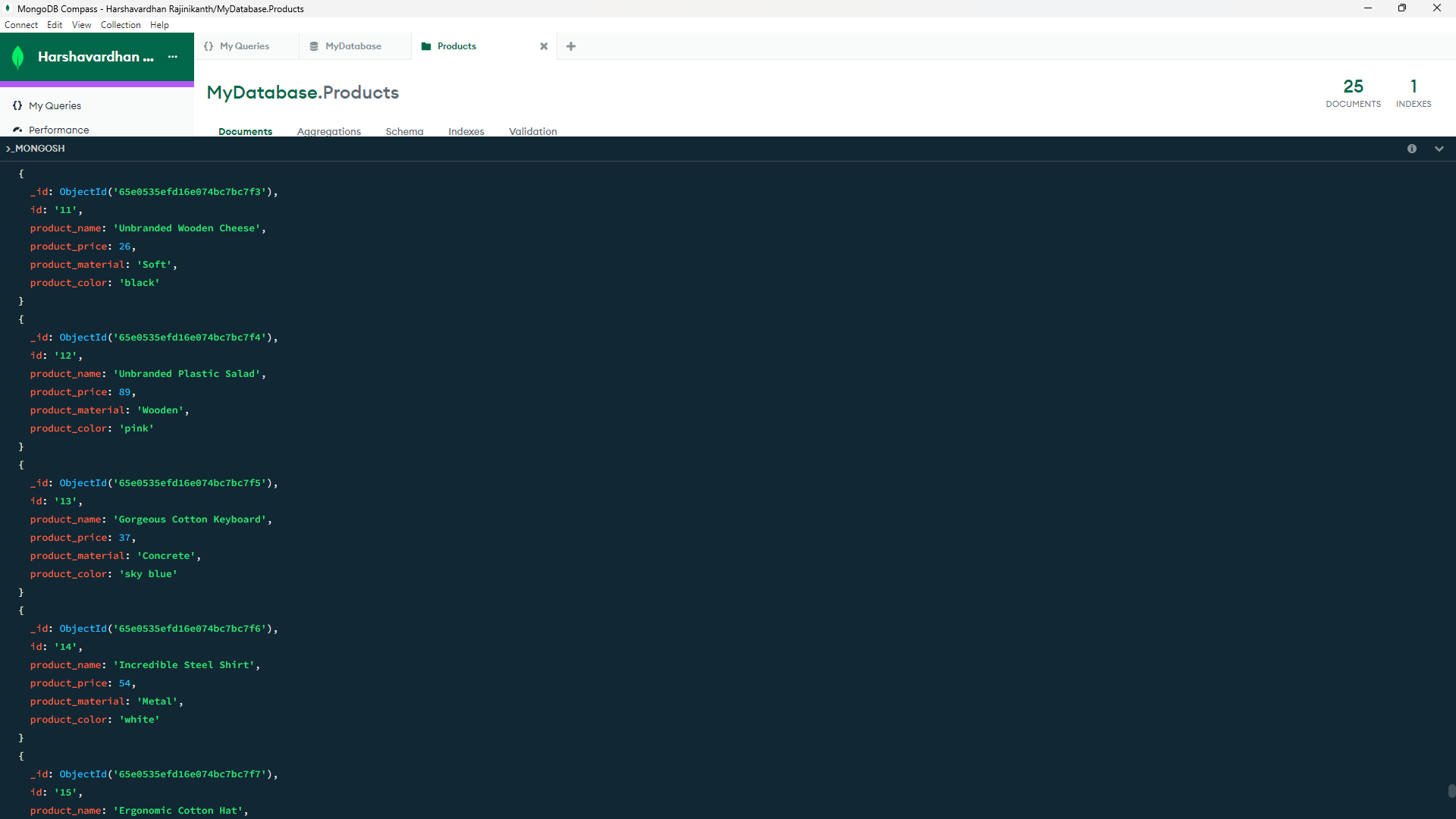
});

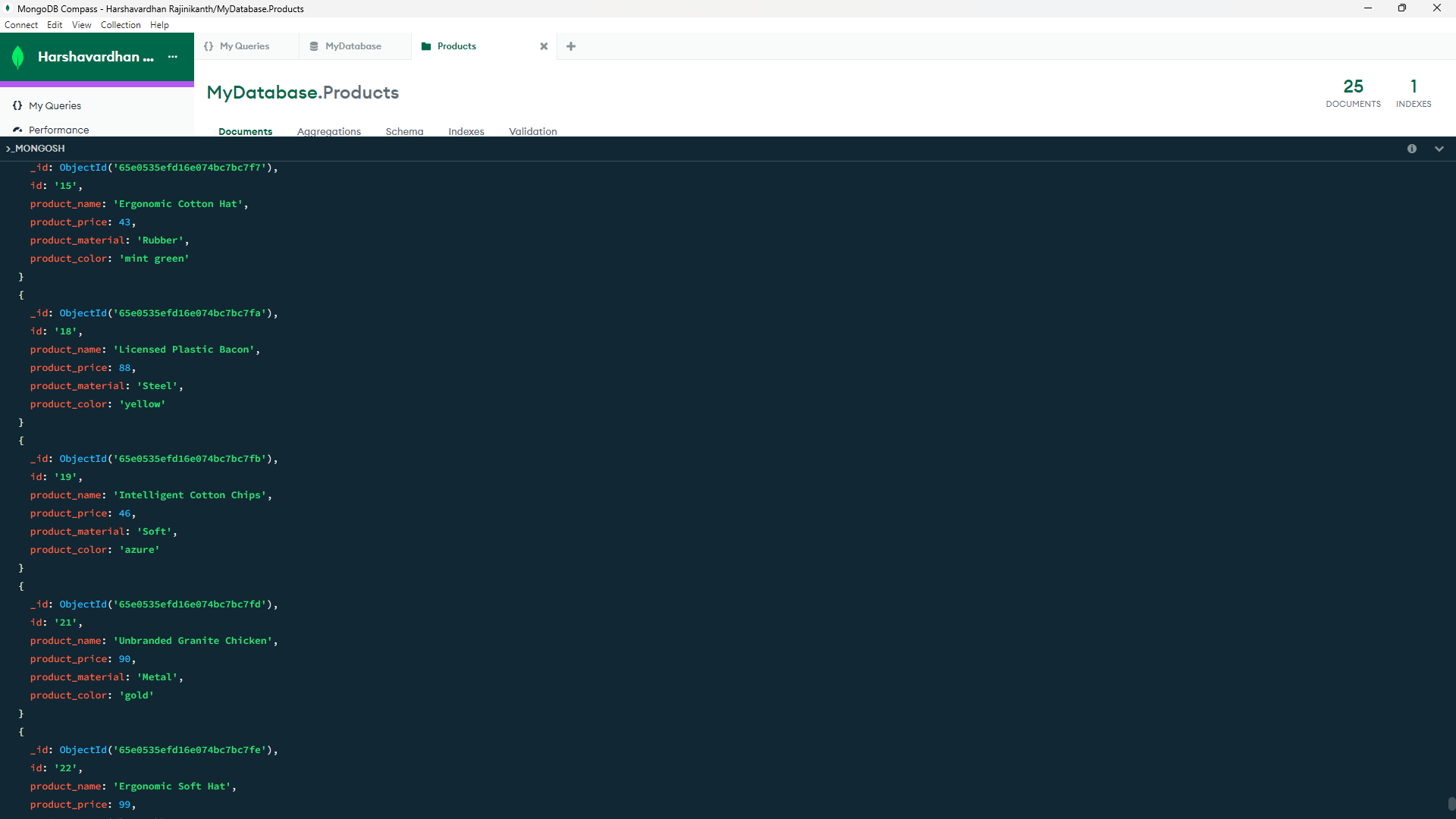
**Output: Deleted Product with ‘id’ are 16,17,20,24**

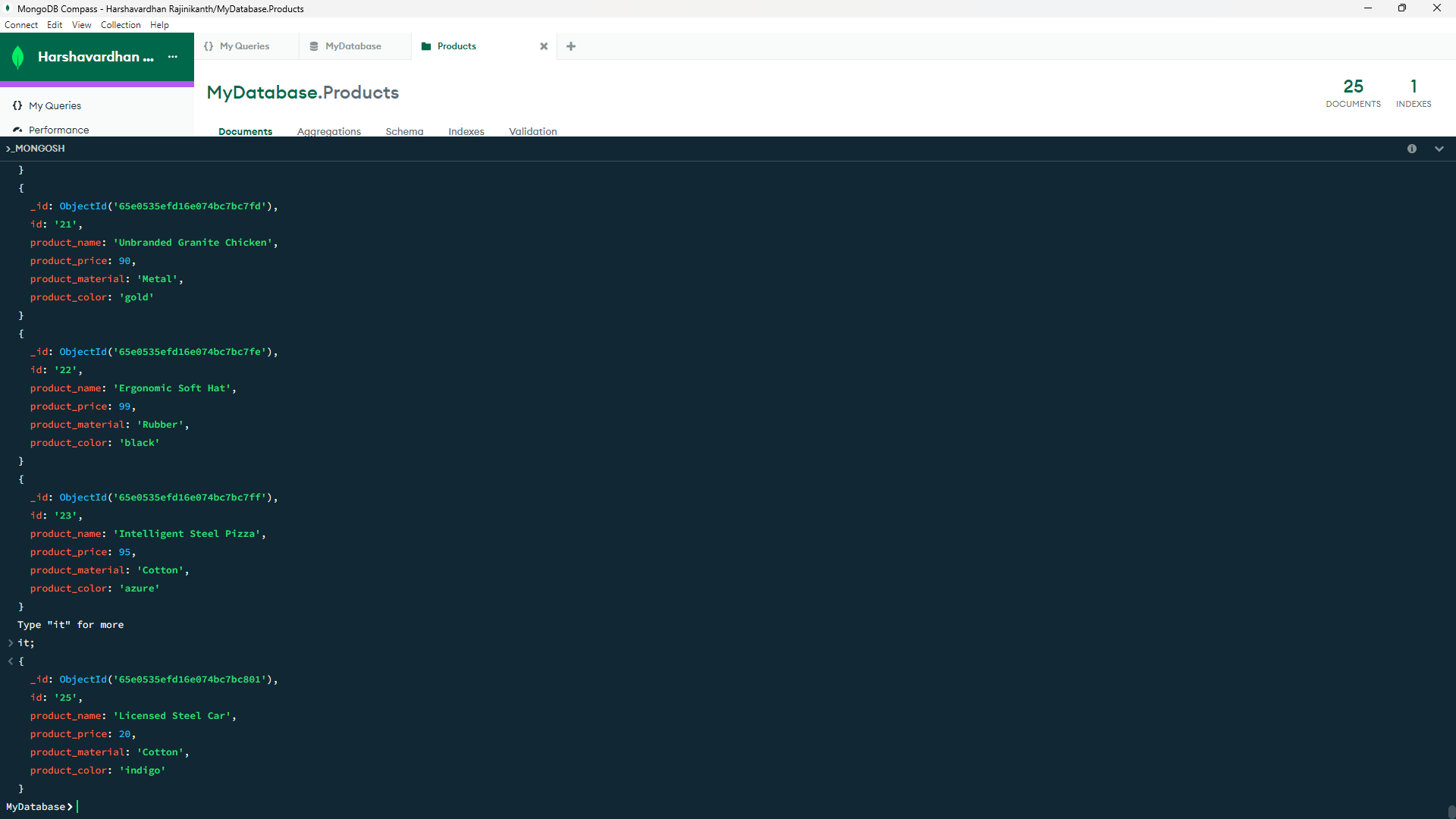
****

****

****

****

****

****