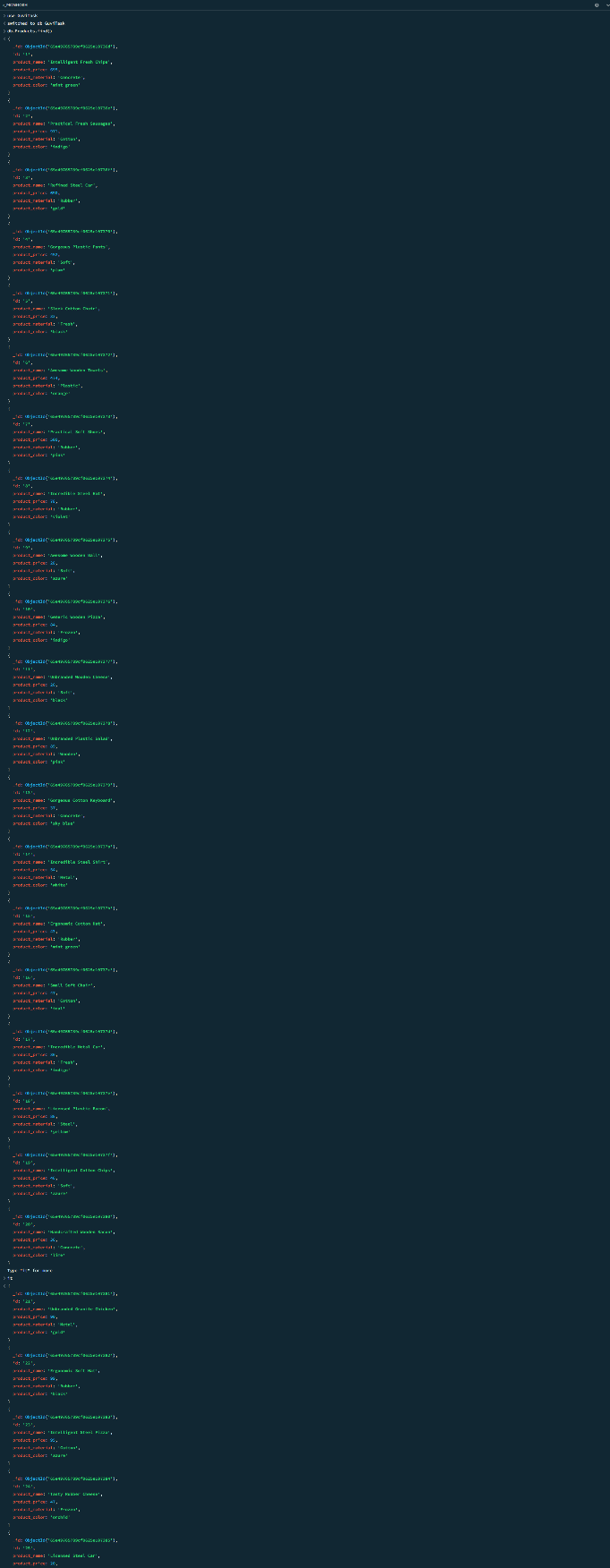
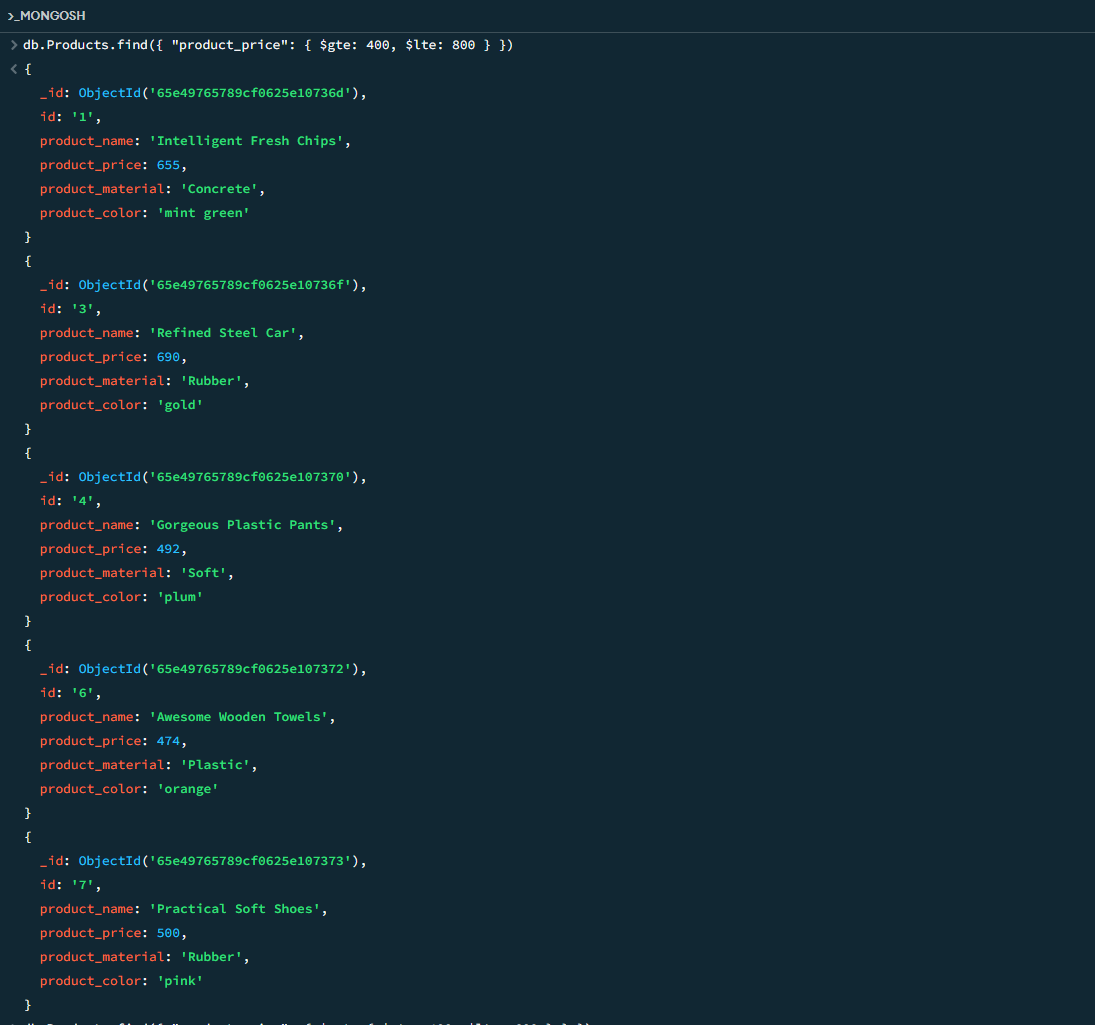
1. **Question**: Find all the information about each products  
   **Answer**: db.Products.find()  
   **Screenshot**:

****

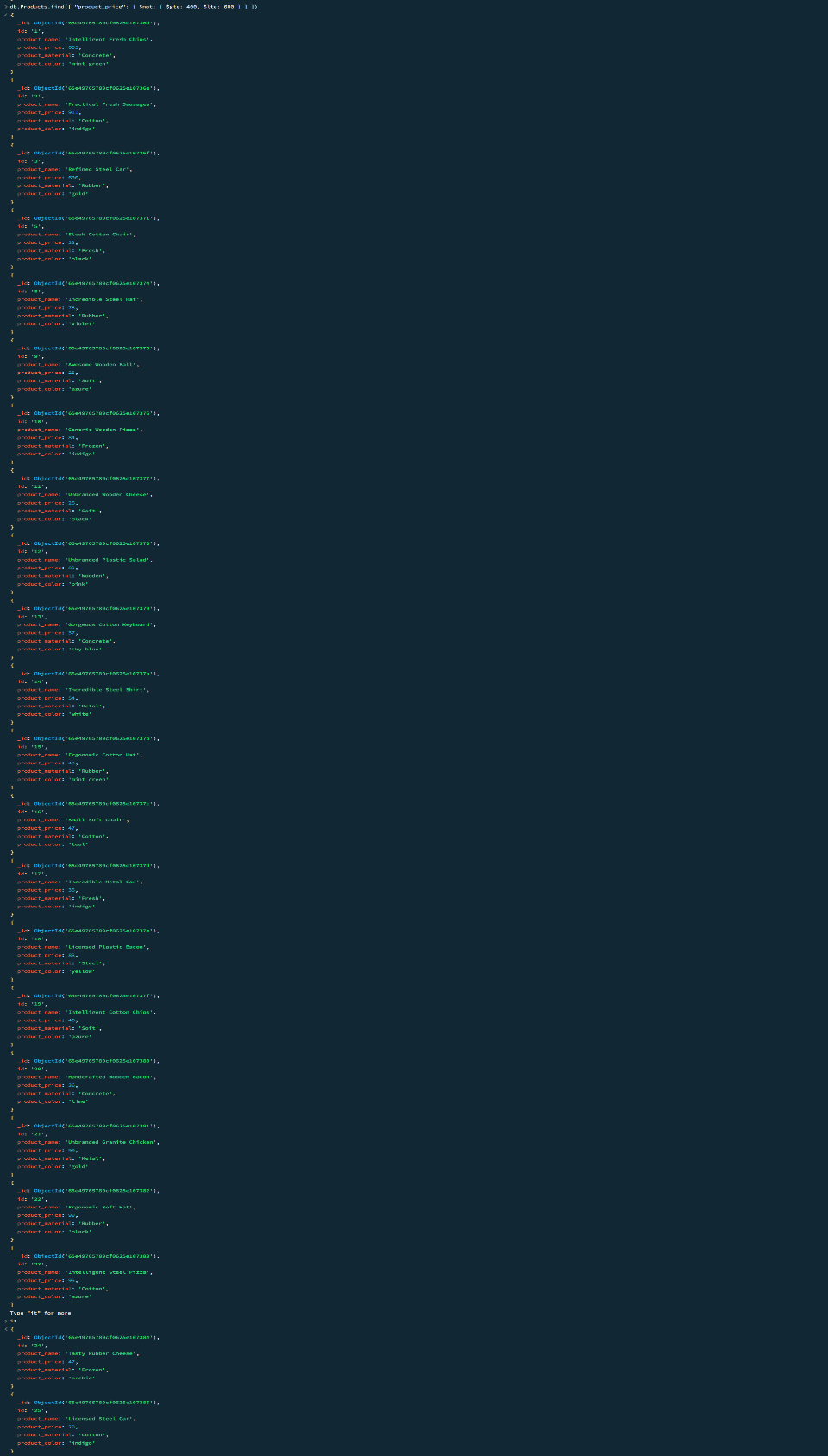
1. **Question**: Find the product price which are between 400 to 800  
   **Answer**: db.Products.find({“product\_price”: {$gte: 400. $lte: 800}})

**Screenshot**:



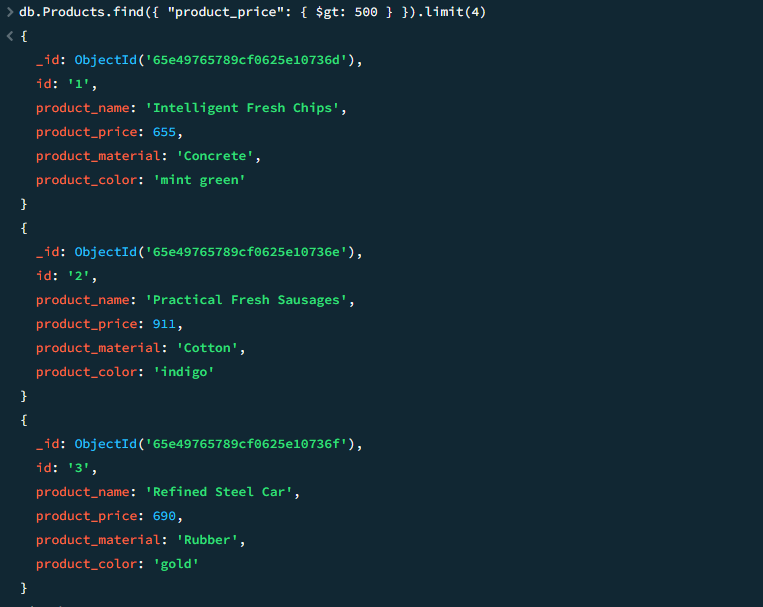
1. **Question**: Find the product price which are not between 400 to 600  
   **Answer**: db.Products.find({“product\_price”: {$not:{$gte: 400, $lte: 600}}})

**Screenshot**:

****

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

**Answer**: db.Products.find({ "product\_price": { $gt: 500 } }).limit(4)

**Screenshot**:

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

**Answer**: db.Products.find({}, { "product\_name": 1, "product\_material": 1, "\_id": 0 })

**Screenshot:**

A screen shot of a computer

Description automatically generated

1. **Question**: Find the product with row id of 10

**Answer**: db.Products.findOne({ "id": "10" })

**Screenshot:**

A computer screen shot of text

Description automatically generated

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

**Answer**: db.Products.find({}, { "product\_name": 1, "product\_material": 1, "\_id": 0 })

**Screenshot:**A screen shot of a computer

Description automatically generated

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

**Answer:** db.Products.find({“product\_material”: {$regex: /soft/i}})

**Screenshot:**

**A screen shot of a computer program

Description automatically generated**

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

**Answer:** db.Products.find({  
$or: [  
{ "product\_color": {$regex: /indigo/i}},   
{"product\_price": {$eq: 492} }  
]  
})

**Screenshot:A blue screen with white text

Description automatically generated**

**Question:** Delete the products which product price value are same  
**Answer:** db.Products.aggregate([

{ $group: { \_id: "$product\_price", count: { $sum: 1 }, ids: { $push: "$\_id" } } },

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

]).forEach(function(doc) {

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

})  
  
**Screenshot: (View next page)***(documents containing same price gets deleted and using* **db.Products.find()** *query we can see the products with different prices only and using* ***db.Products.countDocuments()*** *returns* **21** *)* **A blue screen with white text

Description automatically generated**

**A screen shot of a computer

Description automatically generated**