GUVI DAY 35 (MONGODB DAY1)

NAME : ASHWIN SIVASANKAR V

MongoDB Day 1

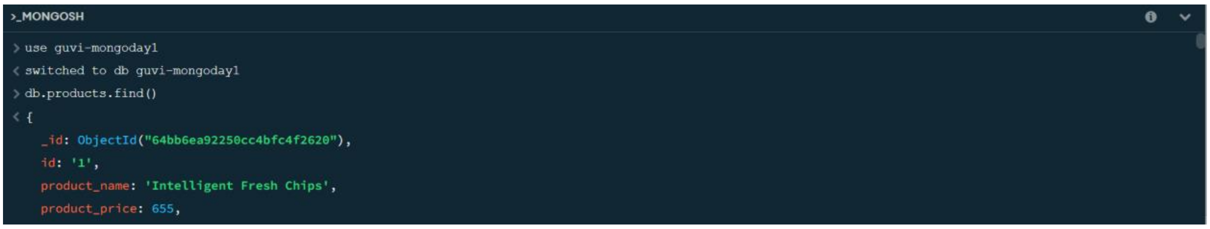
Product JSON: https://github.com/rvsp/database/blob/master/mongodb/product.json

For the following question write the corresponding MongoDB queries

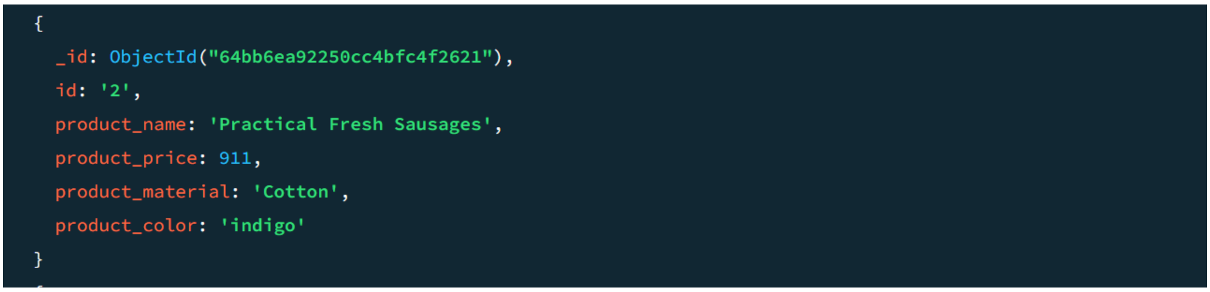
1. Find all the information about each products
2. Find the product price which are between 400 to 800
3. Find the product price which are not between 400 to 600
4. List the four product which are grater than 500 in price
5. Find the product name and product material of each products
6. Find the product with a row id of 10
7. Find only the product name and product material
8. Find all products which contain the value of soft in product material
9. Find products which contain product color indigo and product price 492.00
10. Delete the products which product price value are same

Screenshots:

1.

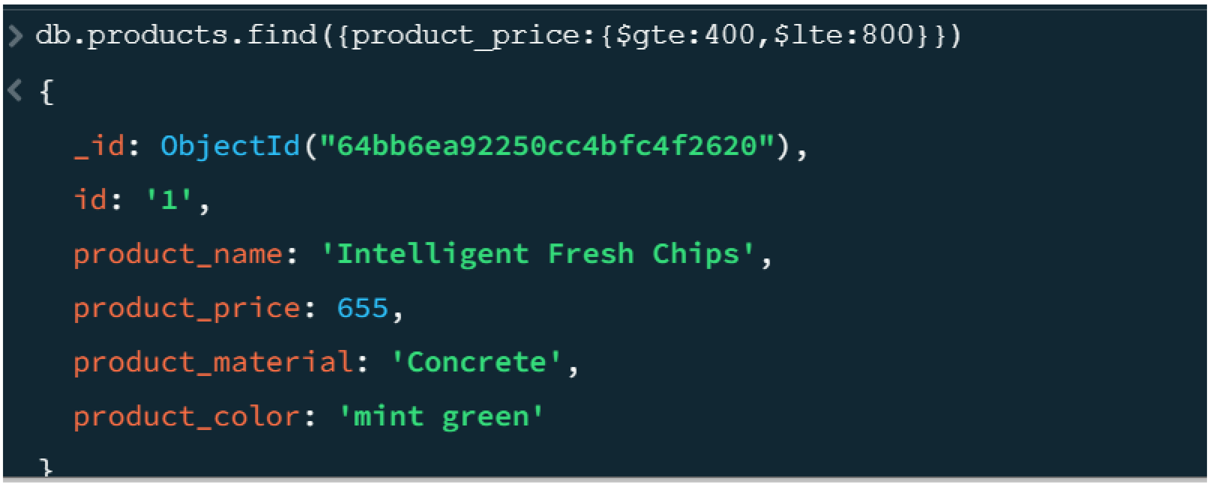


```
> use guvi-mongoday1
< switched to db guvi-mongoday1
> db.products.find()
< {
    _id: ObjectId("64bb6ea92250cc4bfc4f2620"),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
```



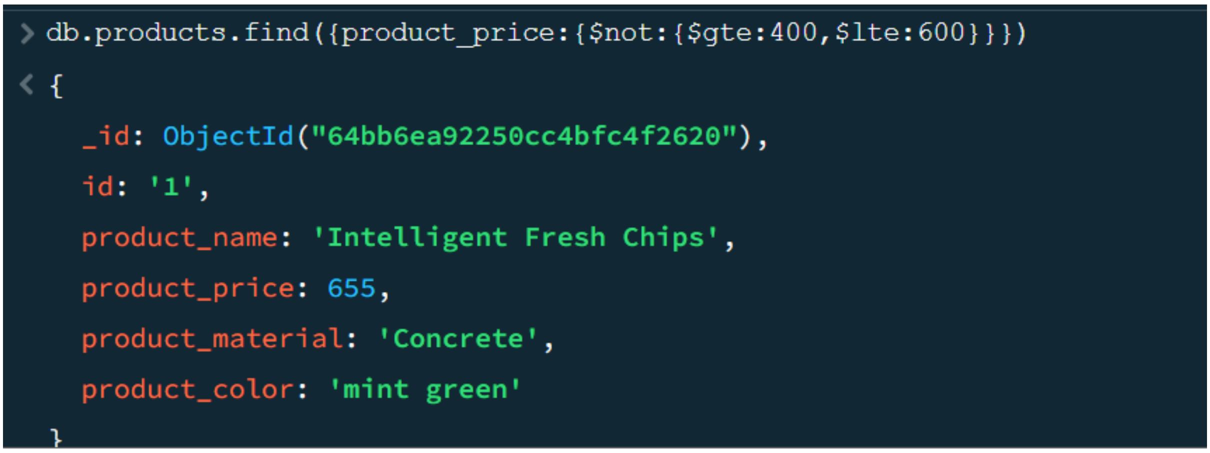
```
  {
    _id: ObjectId("64bb6ea92250cc4bfc4f2621"),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  }
```

2.



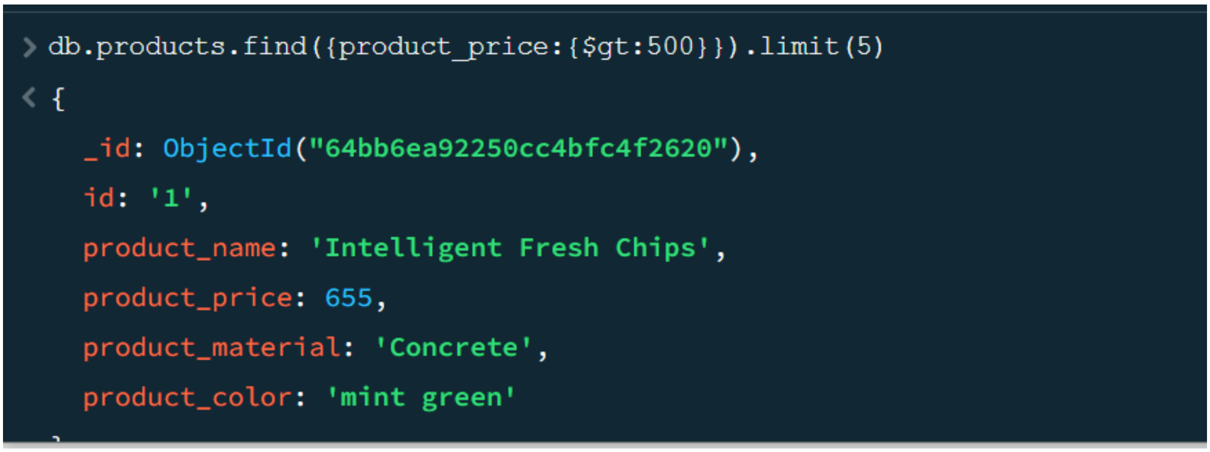
```
> db.products.find({product_price:{$gte:400,$lte:800}})
< {
    _id: ObjectId("64bb6ea92250cc4bfc4f2620"),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  }
```

3.



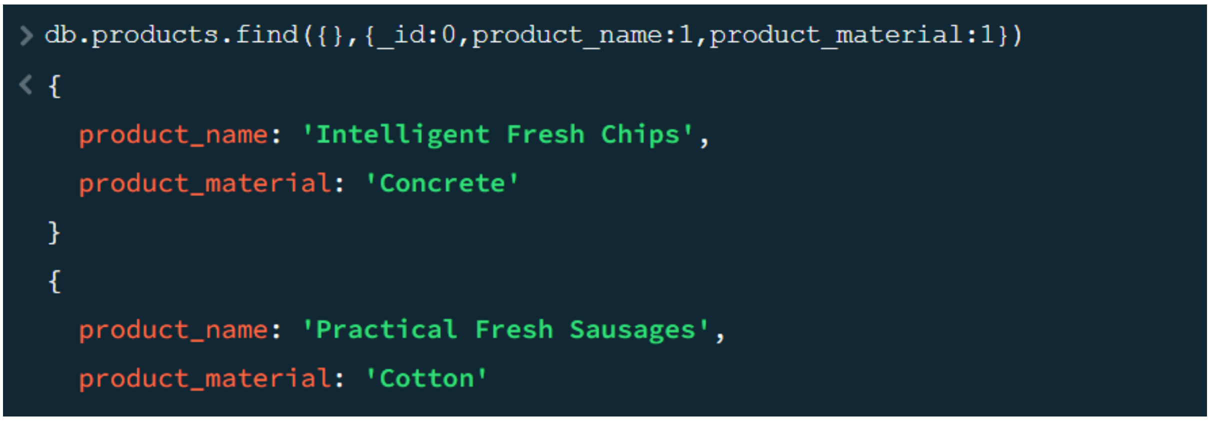
```
> db.products.find({product_price:{$not:{$gte:400,$lte:600}}})
< {
    _id: ObjectId("64bb6ea92250cc4bfc4f2620"),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  }
```

4.



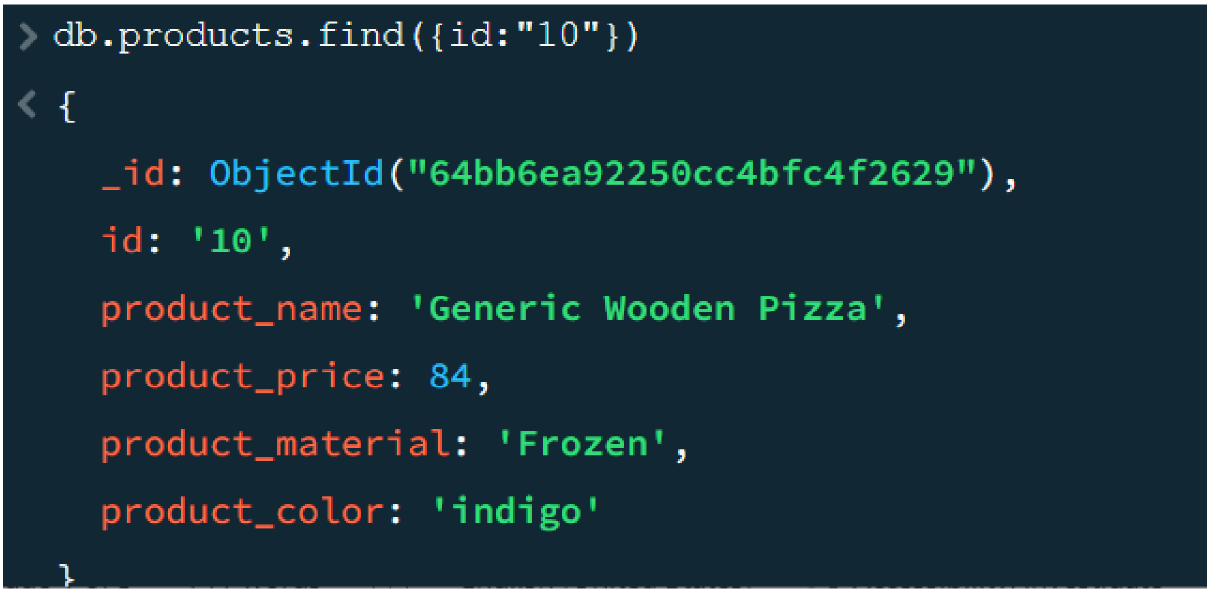
```
> db.products.find({product_price:{$gt:500}}).limit(5)
< {
    _id: ObjectId("64bb6ea92250cc4bfc4f2620"),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
```

5.



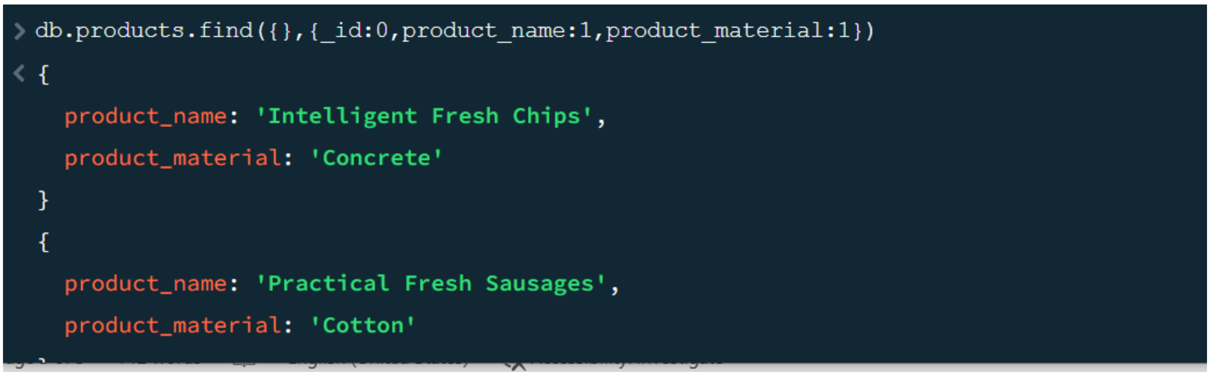
```
> db.products.find({},{_id:0,product_name:1,product_material:1})
< {
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  }
  {
    product_name: 'Practical Fresh Sausages',
    product_material: 'Cotton'
```

6.



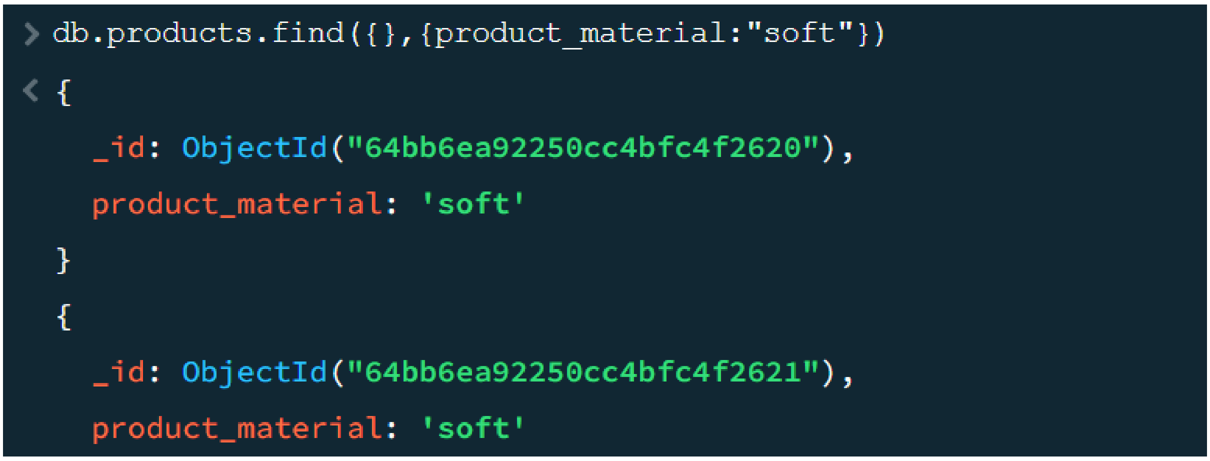
```
> db.products.find({id:"10"})
< {
    _id: ObjectId("64bb6ea92250cc4bfc4f2629"),
    id: '10',
    product_name: 'Generic Wooden Pizza',
    product_price: 84,
    product_material: 'Frozen',
    product_color: 'indigo'
  }
```

7.



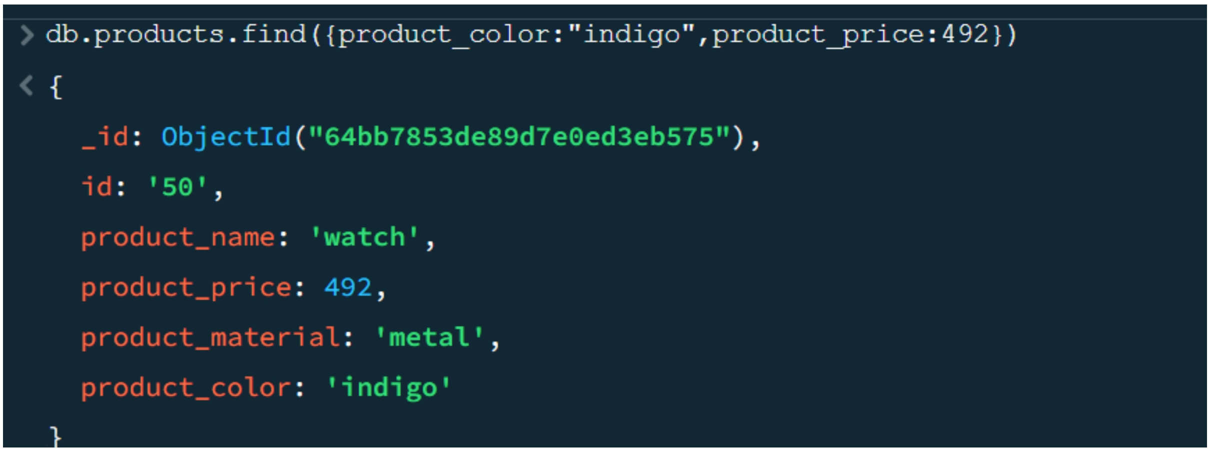
```
> db.products.find({},{_id:0,product_name:1,product_material:1})
< {
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  }
  {
    product_name: 'Practical Fresh Sausages',
    product_material: 'Cotton'
```

8.



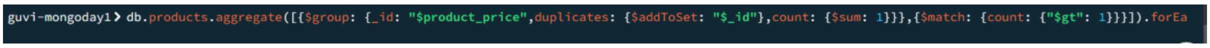
```
> db.products.find({},{product_material:"soft"})
< {
    _id: ObjectId("64bb6ea92250cc4bfc4f2620"),
    product_material: 'soft'
  }
  {
    _id: ObjectId("64bb6ea92250cc4bfc4f2621"),
    product_material: 'soft'
```

9.

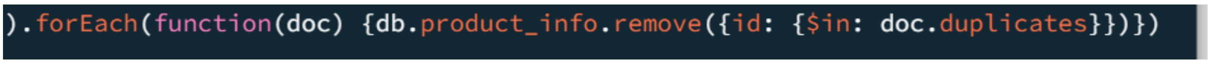


```
> db.products.find({product_color:"indigo",product_price:492})
< {
    _id: ObjectId("64bb7853de89d7e0ed3eb575"),
    id: '50',
    product_name: 'watch',
    product_price: 492,
    product_material: 'metal',
    product_color: 'indigo'
  }
```

10.



```
guvi-mongoday1> db.products.aggregate([{$group: {_id: "$product_price",duplicates: {$addToSet: "$_id"},count: {$sum: 1}}},{$match: {count: {"$gt": 1}}}]).forEa
```



```
).forEach(function(doc) {db.product_info.remove({id: {$in: doc.duplicates}})})
```