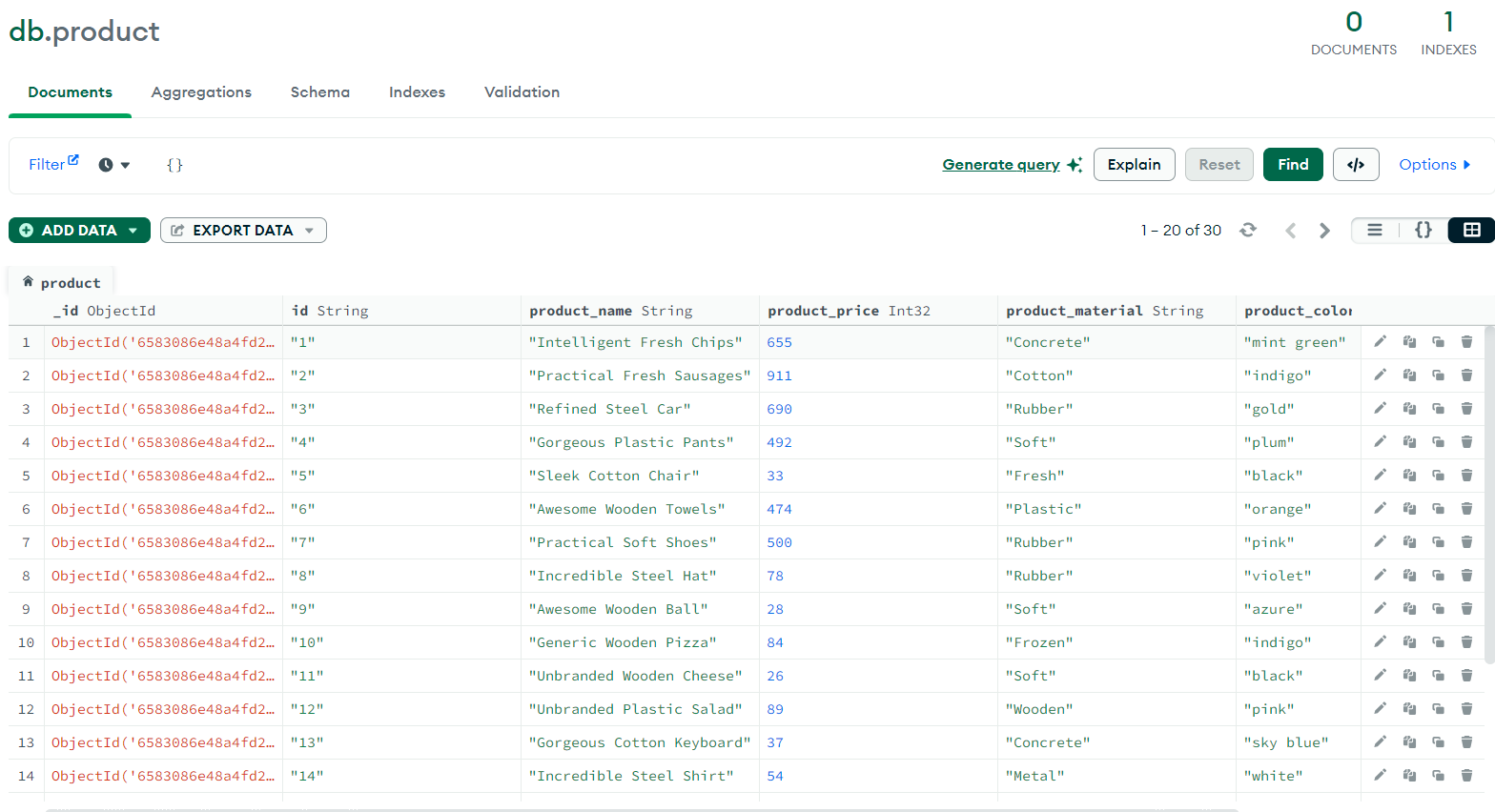
1.Find all the information about each products?

**db.getCollection('product').find({});**

**output:**

****

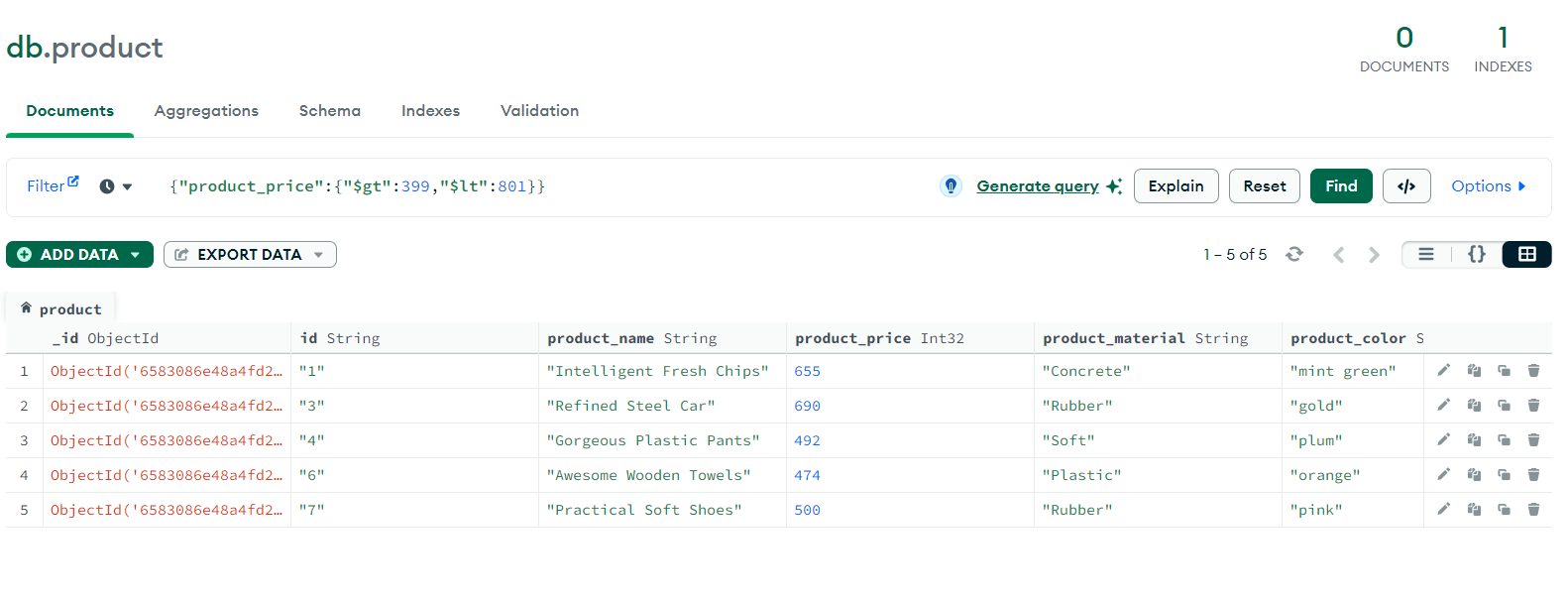
2.Find the product price which are between 400 to 800?

**db.getCollection('product').find({**

**product\_price: { $gt: 399, $lt: 801 }**

**});**

**output:**

****

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

db.getCollection('product').find({

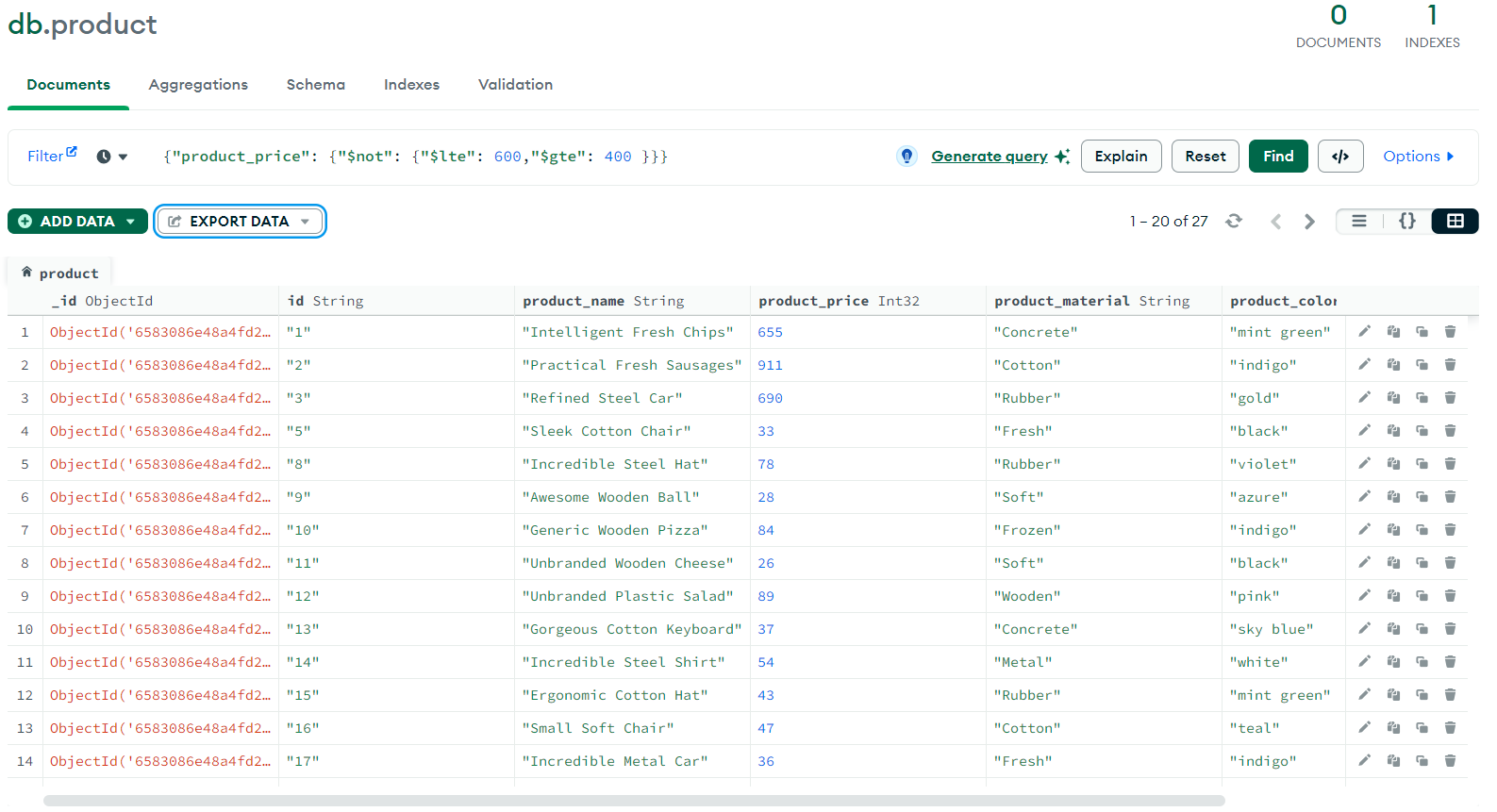
product\_price: {

$not: { $lte: 600, $gte: 400 }

}

});

**output:**

****

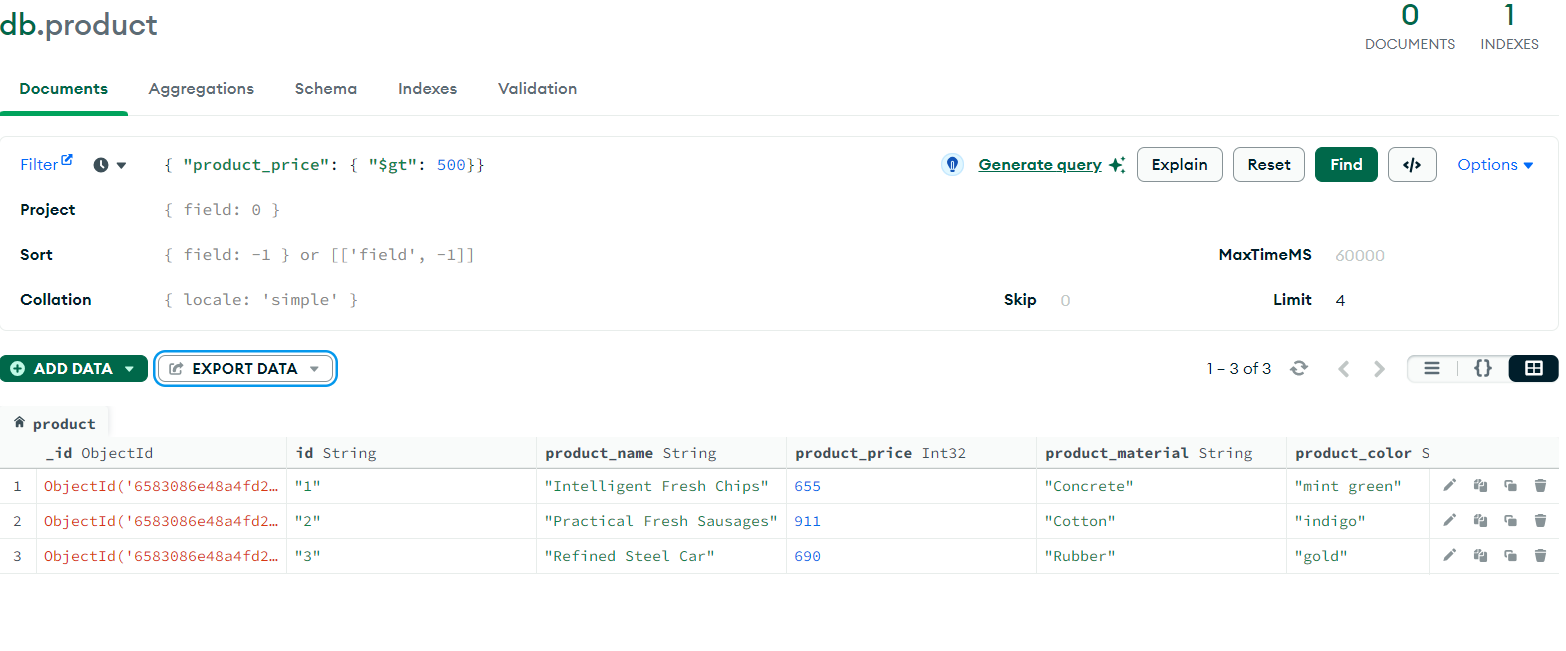
4.List the four products which are greater than 500 in price?

db.getCollection('product')

.find({ product\_price: { $gt: 500 } })

.limit(4);

**output:**



5. Find the product name and product material of each products?

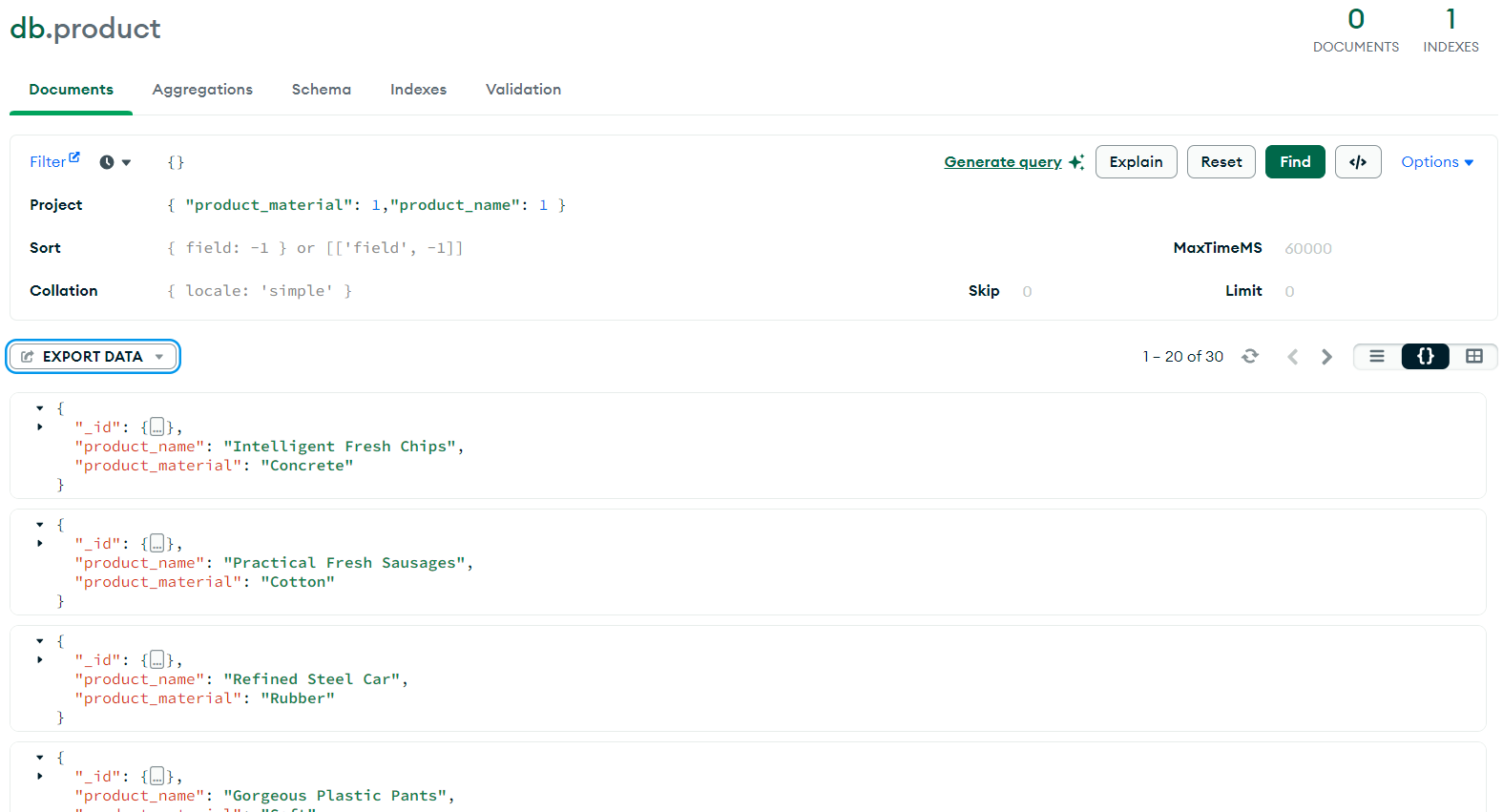
db.getCollection('product').find(

{},

{ product\_material: 1, product\_name: 1 }

);

**output:**



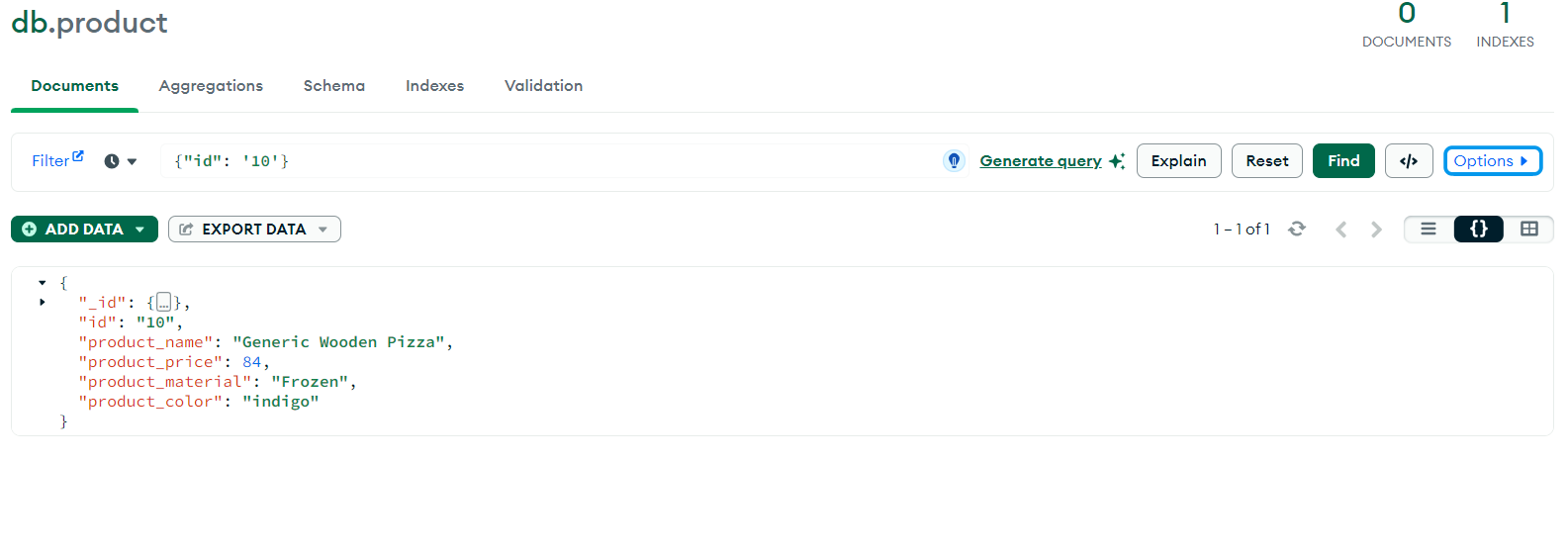
6. Find the product with a row id of 10?

db.getCollection('product').find({ id: '10' },

{}

);

**output:**



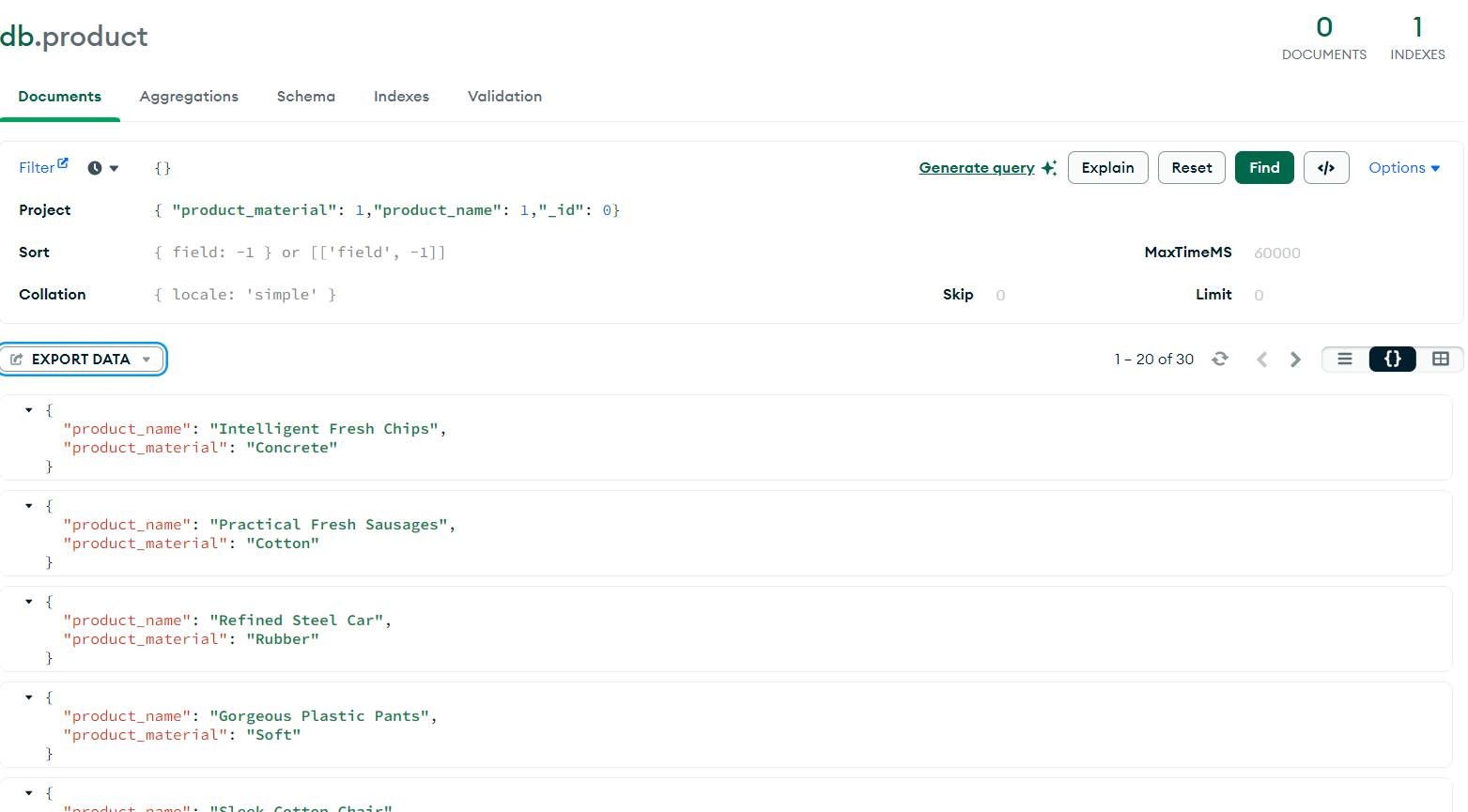
7. Find only the product name and product material?

db.getCollection('product').find({},

{ product\_material: 1, product\_name: 1, \_id: 0 }

);

**output:**

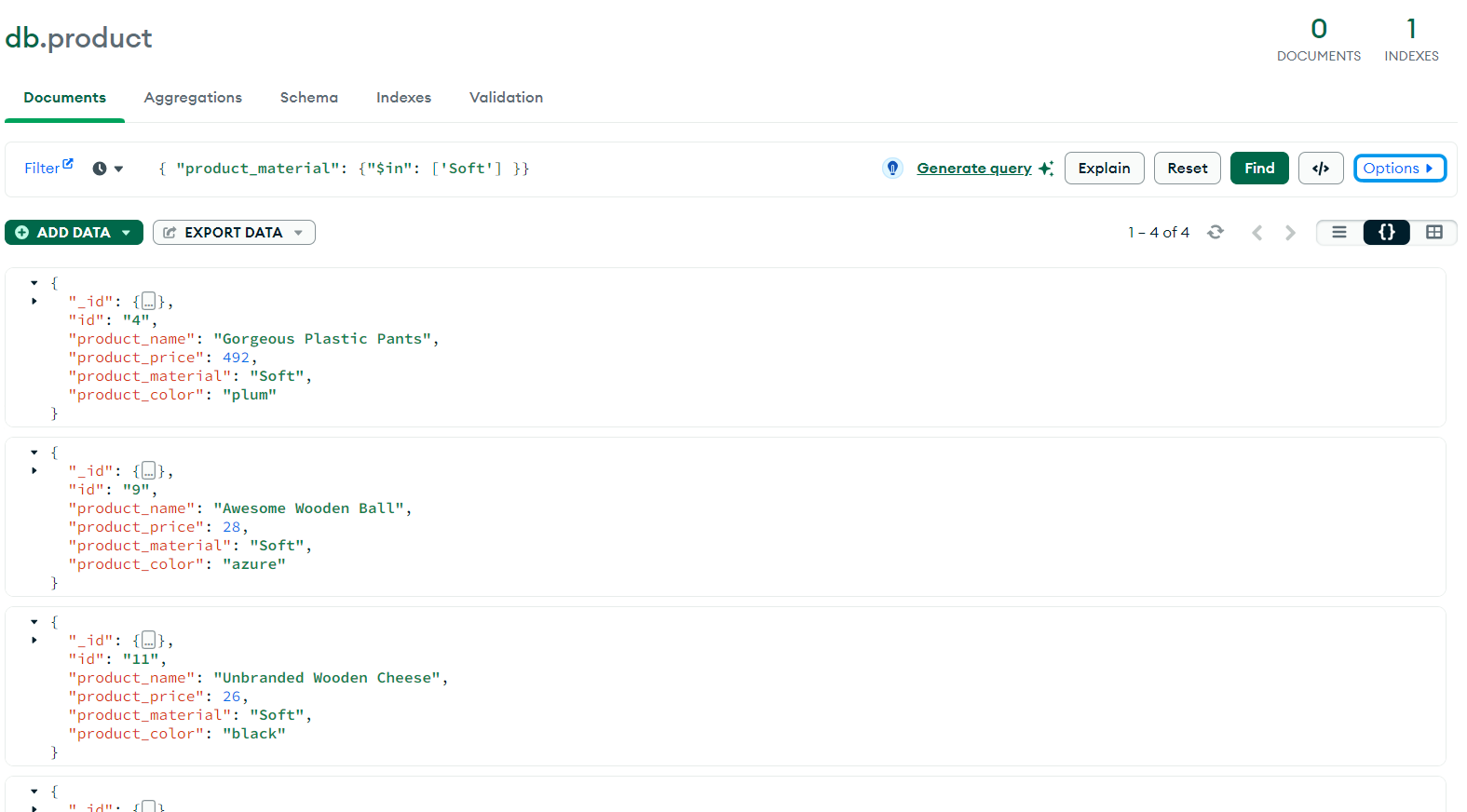


8. Find all products which contain the value of soft in product material?

db.getCollection('product').find({ product\_material: { $in: ['Soft'] } },

{});

**output:**

****

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

db.getCollection('product').find({

$or: [

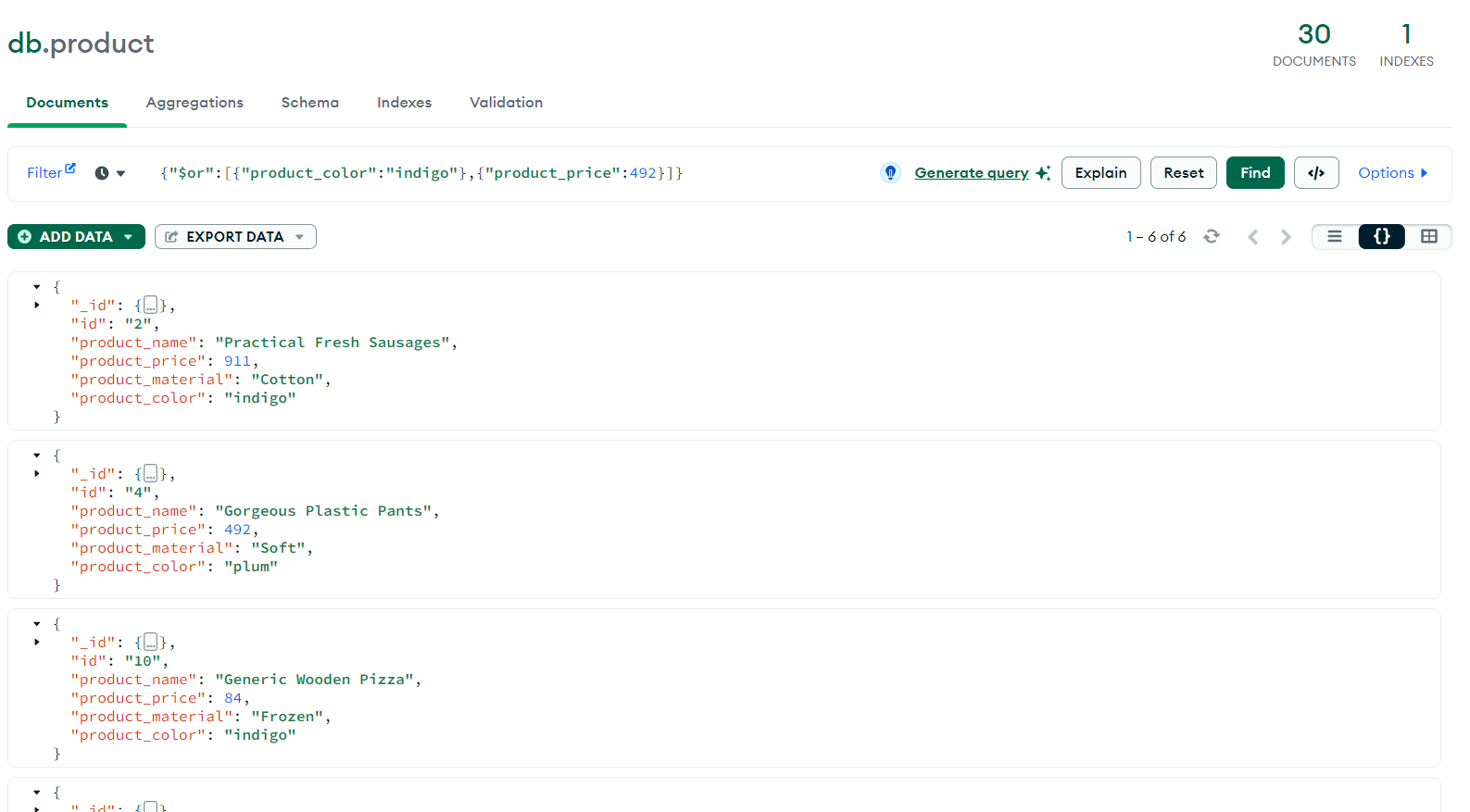
{ product\_color: 'indigo' },

{ product\_price: 492 }

]

});

**output:**

****

10. Delete the products which product price value are same?

db.getCollection('product'). aggregate([{ '$group': {

'\_id': '$product\_price',

'ids': {

'$push': '$id'

},

'count': {'$sum' }

}

}, {

'$match': {

'count': { '$gt': 1 }

}

}

].forEach(function(doc){db.product.deleteone

({id:item.ids[0]});});

**Output**:

