MongoDB Day 1

Database Name: GUVI-FSD-TASK
Collection Name: product

For the following question write the corresponding MongoDB queries

1. Find all the information about each products
   Ans: db.product.find();

2. Find the product price which are between 400 to 800
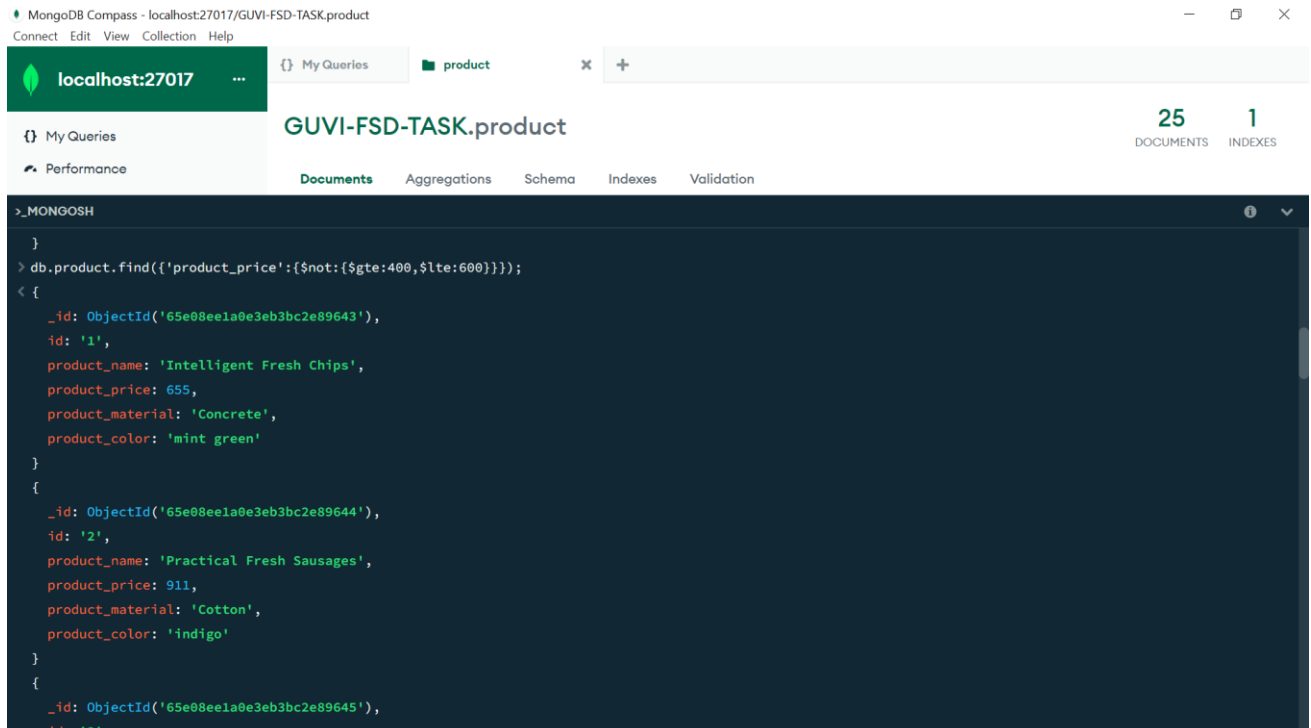   Ans: db.product.find({'product_price':{$gte:400,$lte:800}});

3. Find the product price which are not between 400 to 600
    Ans: db.product.find({'product_price':{$not:{$gte:400,$lte:600}}});

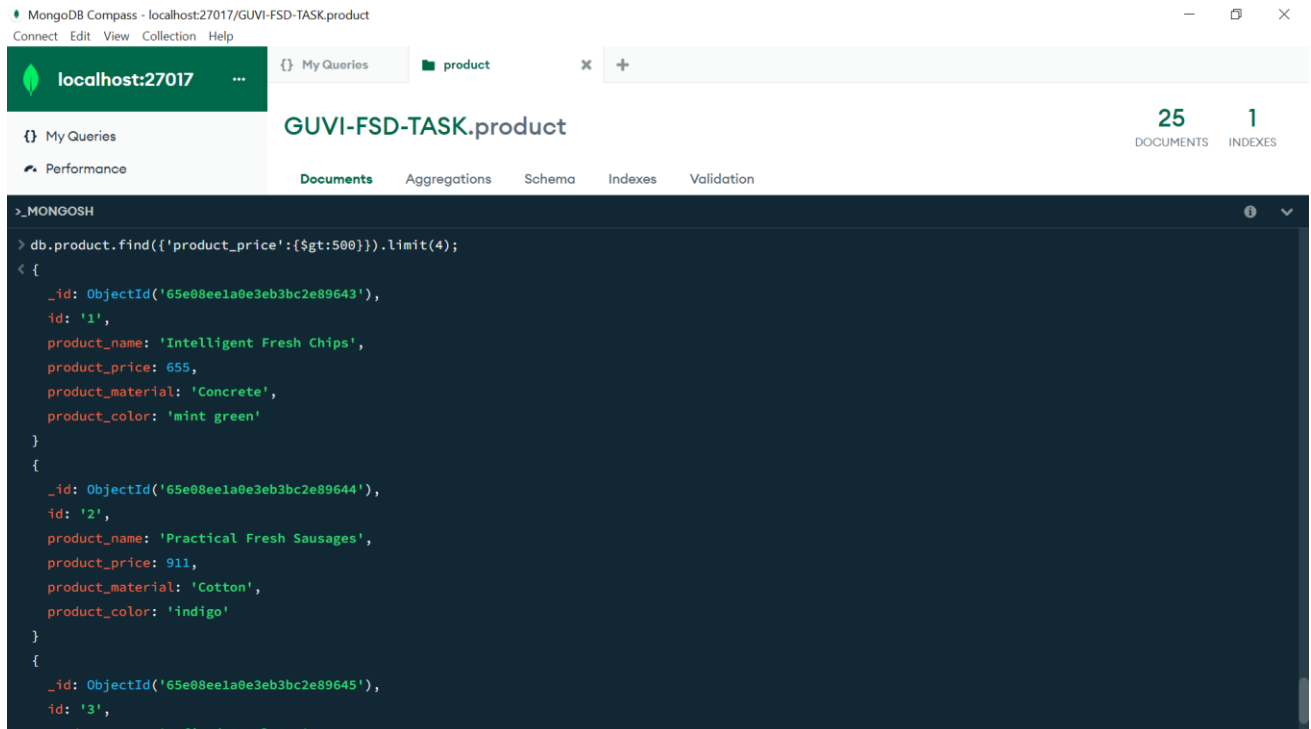4. List the four product which are greater than 500 in price
   Ans: db.product.find({'product_price':{$gt:500}}).limit(4);


```
MongoDB Compass - localhost:27017/GUVI-FSD-TASK.product
Connect  Edit  View  Collection  Help

localhost:27017  ...            {} My Queries     product       ×   +

{} My Queries                   GUVI-FSD-TASK.product                                    25          1
Performance                                                                          DOCUMENTS    INDEXES
                                Documents   Aggregations   Schema   Indexes   Validation

>_MONGOSH

> db.product.find({'product_price':{$gt:500}}).limit(4);
< {
    _id: ObjectId('65e08ee1a0e3eb3bc2e89643'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  }
  {
    _id: ObjectId('65e08ee1a0e3eb3bc2e89644'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  }
  {
    _id: ObjectId('65e08ee1a0e3eb3bc2e89645'),
    id: '3',
    product name: 'Refined Steel Car'
```
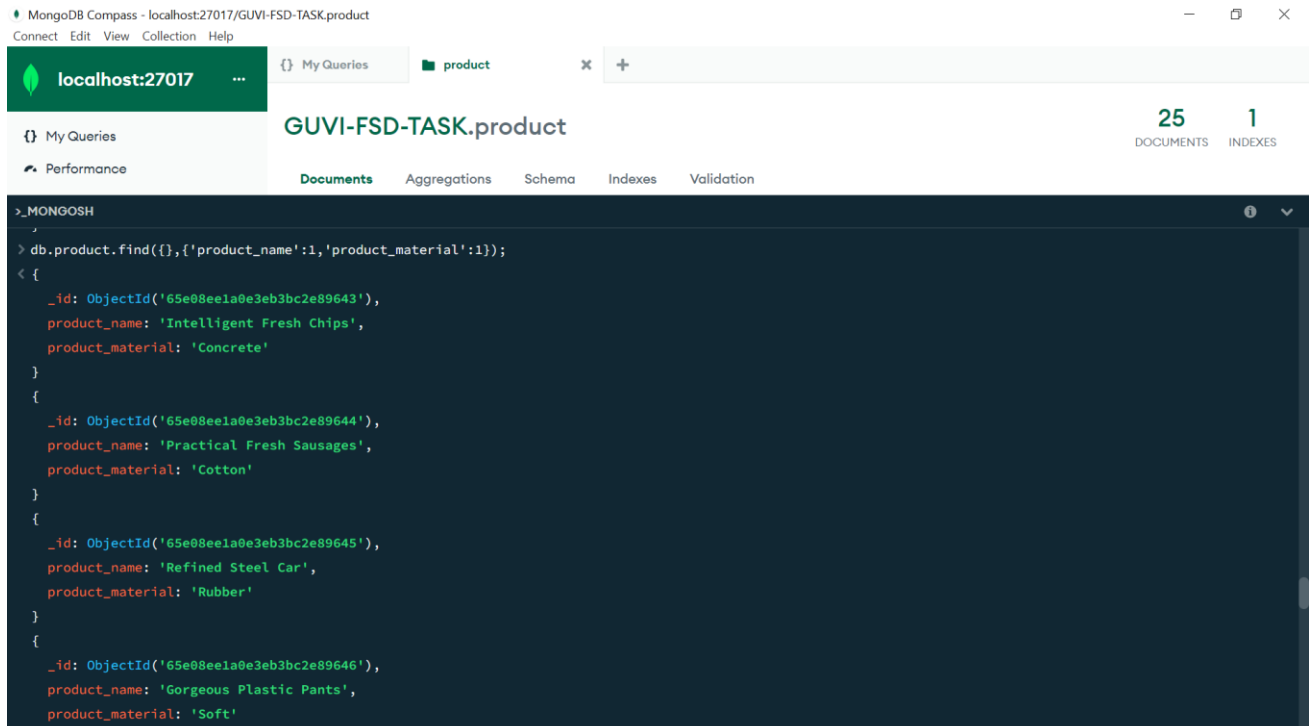
5. Find the product name and product material of each products
   Ans: db.product.find({},{'product_name':1,'product_material':1});


```
MongoDB Compass - localhost:27017/GUVI-FSD-TASK.product
Connect  Edit  View  Collection  Help

localhost:27017  ...            {} My Queries     product       ×   +

{} My Queries                   GUVI-FSD-TASK.product                                    25          1
Performance                                                                          DOCUMENTS    INDEXES
                                Documents   Aggregations   Schema   Indexes   Validation

>_MONGOSH

> db.product.find({},{'product_name':1,'product_material':1});
< {
    _id: ObjectId('65e08ee1a0e3eb3bc2e89643'),
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  }
  {
    _id: ObjectId('65e08ee1a0e3eb3bc2e89644'),
    product_name: 'Practical Fresh Sausages',
    product_material: 'Cotton'
  }
  {
    _id: ObjectId('65e08ee1a0e3eb3bc2e89645'),
    product_name: 'Refined Steel Car',
    product_material: 'Rubber'
  }
  {
    _id: ObjectId('65e08ee1a0e3eb3bc2e89646'),
    product_name: 'Gorgeous Plastic Pants',
    product_material: 'Soft'
```
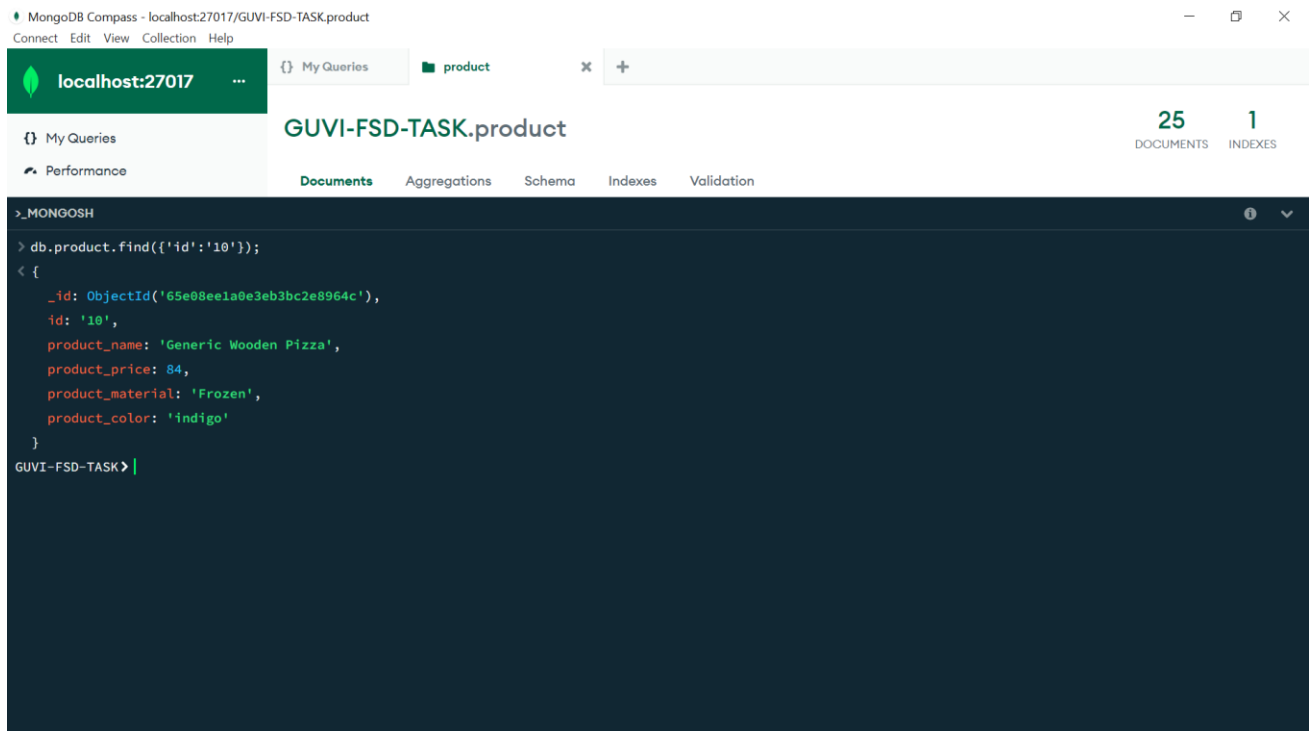
6. Find the product with a row id of 10
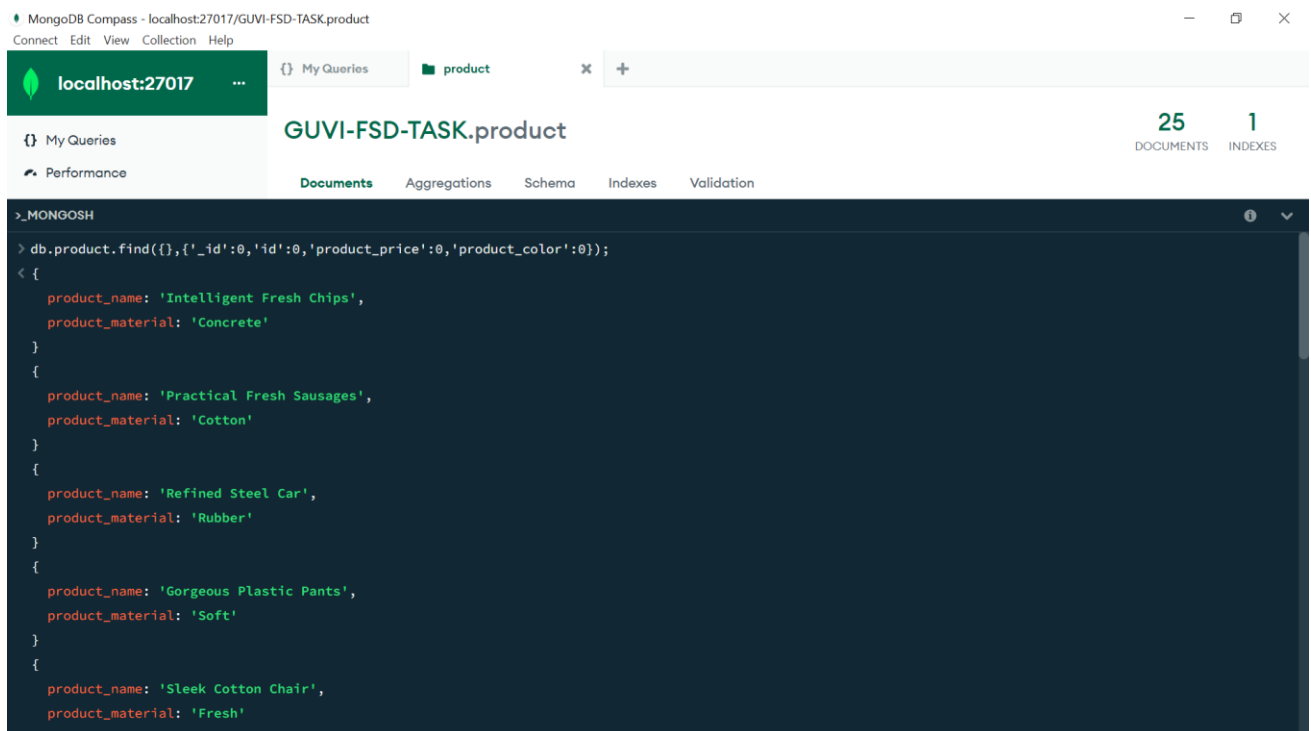   Ans: db.product.find({'id':'10'});

7. Find only the product name and product material
   Ans: db.product.find({},{'_id':0,'id':0,'product_price':0,'product_color':0});

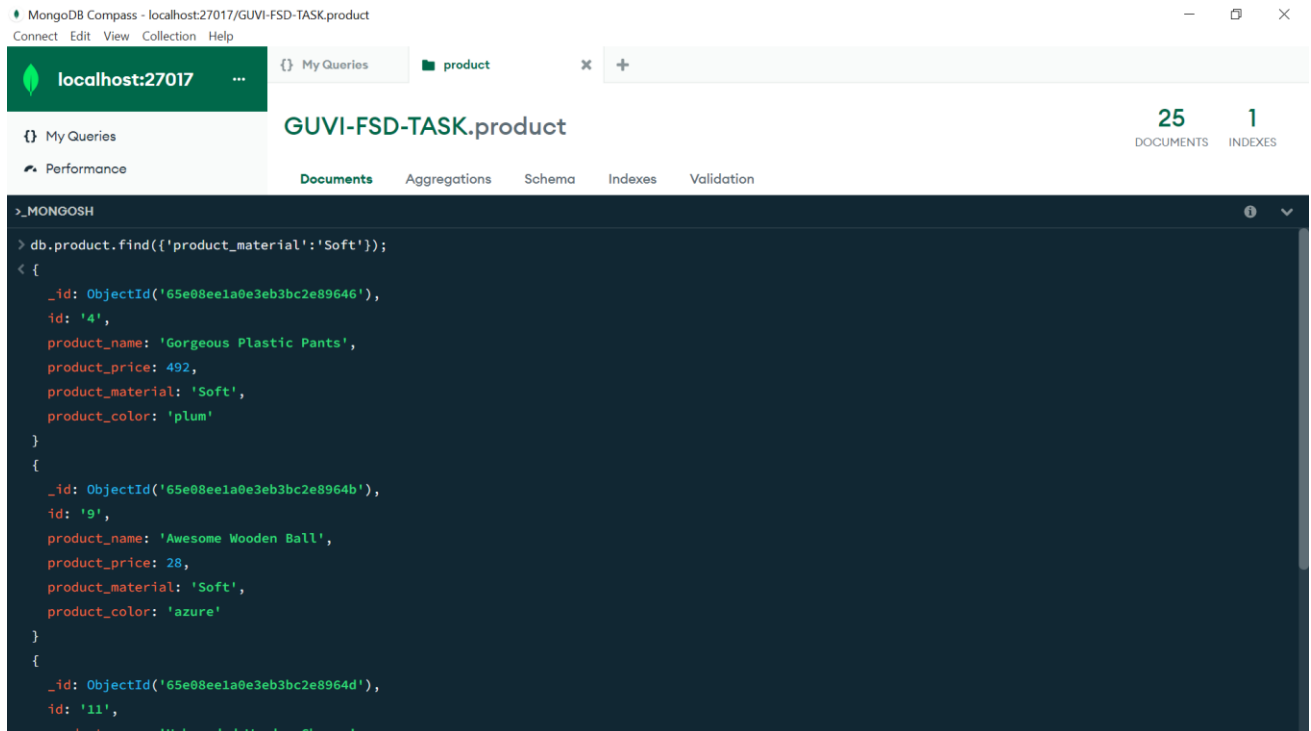8. Find all products which contain the value of soft in product Material
   Ans: db.product.find({'product_material':'Soft'});

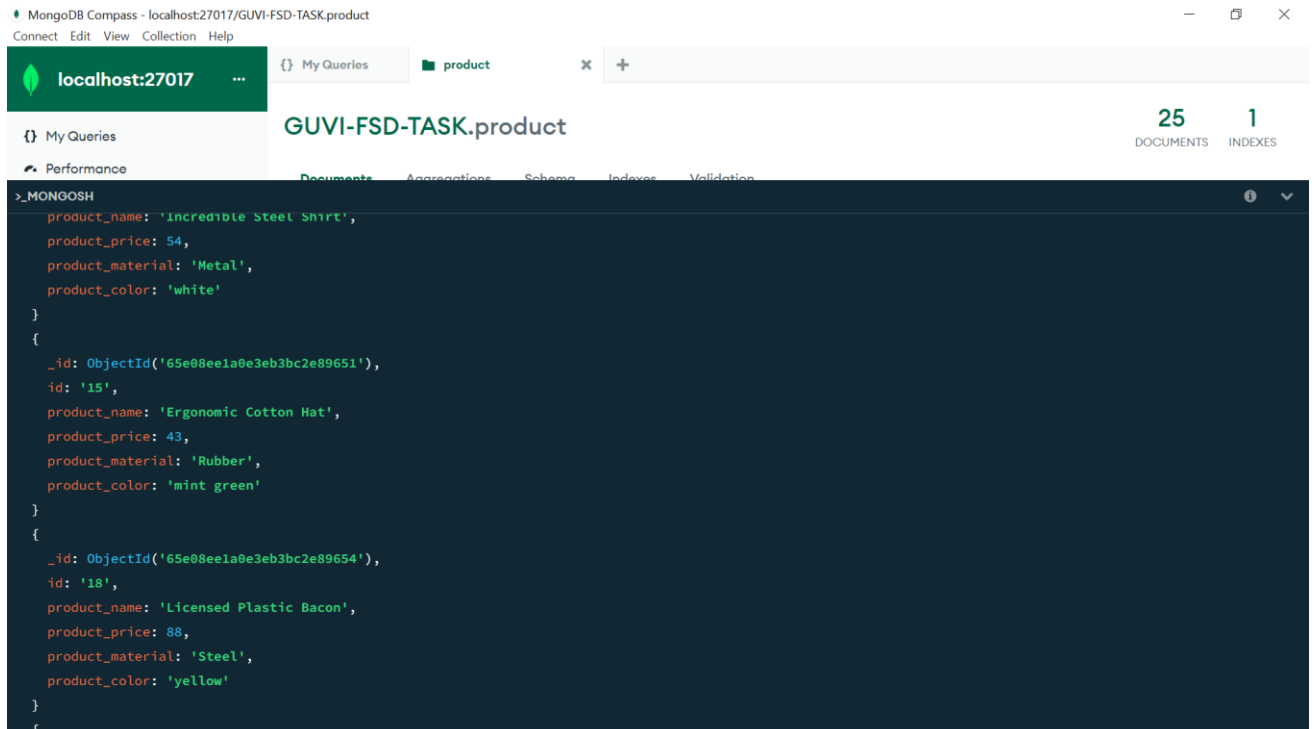9. Find products which contain product color indigo and product price 492.00
   Ans: db.product.find({$and:[{'product_color':'indigo'},{'product_price':492}]});

   Output: No products are match for the given condition.

10. Delete the products which product price value are same
    Ans: db.Products.aggregate([{ $group: {_id: '$product_price',
        count: { $sum: 1 } } }, { $match: { count: { $gt: 1 } } }]).forEach((e)
        => { db.Products.deleteMany({ product_price: e._id });});

    Output: Product with 'id' 16, 17, 20 and 24 are deleted.