MongoDB–Task–1

1) Find all the information about each products.

```
> db.products.find();
```


```
< {
    _id: ObjectId('65e0729787bbd0f590e22c79'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  }
  {
    _id: ObjectId('65e0729787bbd0f590e22c7a'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  }
  {
    _id: ObjectId('65e0729787bbd0f590e22c7b'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  }
```


Totally got **25 Document** of informations.

2) Find the product price which are between 400 to 800

```
> db.products.find({"product_price":{$gt:400, $lt:800}});;
```

```
{
  _id: ObjectId('65e0729787bbd0f590e22c79'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('65e0729787bbd0f590e22c7b'),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
  product_material: 'Rubber',
  product_color: 'gold'
}
{
  _id: ObjectId('65e0729787bbd0f590e22c7c'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
}
```

Totally **5 Documents** Have price between 400 to 800.

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

```
> db.products.find({"product_price":{$not:{$gt:400,$lt:600}}});
```

```
< {
    _id: ObjectId('65e0729787bbd0f590e22c79'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  }
  {
    _id: ObjectId('65e0729787bbd0f590e22c7a'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  }
  {
    _id: ObjectId('65e0729787bbd0f590e22c7b'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  }
```

```
> db.products.find({"product_price":{$not:{$gt:400,$lt:600}}}).count();
< 22
```

Totally **22 Documents** not between 400 to 600.

4) List the four product which are greater than 500 in price

```
> db.products.find({'product_price':{$gt:500}}).limit(4);
```

Totally we got only **3 Documents** have product price greater than 500.

5) Find the product name and product material of each products

```
> 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'
  }
  {
    product_name: 'Refined Steel Car',
    product_material: 'Rubber'
  }
  {
    product_name: 'Gorgeous Plastic Pants',
    product_material: 'Soft'
  }
  {
    product_name: 'Sleek Cotton Chair',
    product_material: 'Fresh'
  }
  {
    product_name: 'Awesome Wooden Towels',
    product_material: 'Plastic'
  }
```

```
> db.products.find({},{'_id':0,'product_name':1,'product_material':1}).count();
< 25
```

We got **25 components** product name and product material in result .

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

```
> db.products.find({'id':'10'});
```

```
< {
    _id: ObjectId('65e0729787bbd0f590e22c82'),
    id: '10',
    product_name: 'Generic Wooden Pizza',
    product_price: 84,
    product_material: 'Frozen',
    product_color: 'indigo'
  }
```

Only **one product Document** have an row if 10.

---

7) Find only the product name and product material.

```
> db.products.find({'id':'8'},{'_id':0,'product_name':1,'product_material':1});
```

```
< {
    product_name: 'Incredible Steel Hat',
    product_material: 'Rubber'
  }
```

Randomly I choose an id:8 ‘s product name and product material.

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

```
> db.products.find({'product_material':'Soft'});
```

```
< {
    _id: ObjectId('65e0729787bbd0f590e22c7c'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  }
  {
    _id: ObjectId('65e0729787bbd0f590e22c81'),
    id: '9',
    product_name: 'Awesome Wooden Ball',
    product_price: 28,
    product_material: 'Soft',
    product_color: 'azure'
  }
  {
    _id: ObjectId('65e0729787bbd0f590e22c83'),
    id: '11',
    product_name: 'Unbranded Wooden Cheese',
    product_price: 26,
    product_material: 'Soft',
    product_color: 'black'
  }
```

```
> db.products.find({'product_material':'Soft'}).count();
< 4
```

Totally, **4 Product document** contain product material as Soft.

9) Find products which contain product colour indigo and product price 492.00.

```
> db.products.find({$and:[{'product_color':'indigo'},{'product_price':492.00}]});
<
MongoDB-Task1>
```

There is no, Product have a Document with product colour indigo and product price 492.00

---

10) Delete the products which product price value are same

```
> db.products.aggregate([{$group:{_id:"$product_price",duplicate:{$addToSet:"$_id"},
            count:{$sum:1}}},
            {$match:{count:{$gt:1}
            }
          }
       ]).forEach(function(doc){db.product.deleteOne({_id:{$in:doc.duplicates}})
     })
```

There is no products contain same product price.