

1. Find all the information about each products

Query:

```
> db.products.find().toArray();
< [
  {
    _id: ObjectId('65aa8876d792947f8dbaf956'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('65aa8876d792947f8dbaf957'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('65aa8876d792947f8dbaf958'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  ...
]
```

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

Query:

```
> db.products.find({ $and: [{product_price: { $gt: 400 }}, {product_price: { $lt: 800 }} ] }).toArray()
< [
  {
    _id: ObjectId('65aa8876d792947f8dbaf956'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('65aa8876d792947f8dbaf958'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {
    _id: ObjectId('65aa8876d792947f8dbaf959'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
]
```

- Find the product price which are not between 400 to 600

Query:

```
>_MONGOSH
> db.products.find({$nor:[{product_price: {$gt: 400, $lt: 600}}]}).toArray();
< [
  {
    _id: ObjectId('65aa8876d792947f8dbaf956'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('65aa8876d792947f8dbaf95e'),
    id: '9',
    product_name: 'Awesome Wooden Ball',
    product_price: 28,
    product_material: 'Soft',
    product_color: 'azure'
  },
  {
    _id: ObjectId('65aa8876d792947f8dbaf960'),
    id: '11',
    product_name: 'Unbranded Wooden Cheese',
    product_price: 26,
    product_material: 'Soft',
    product_color: 'black'
  },
]
```

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

>_MONGOSH

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

```
< {
  _id: ObjectId('65aa8876d792947f8dbaf956'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('65aa8876d792947f8dbaf958'),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
  product_material: 'Rubber',
  product_color: 'gold'
}
{
  _id: ObjectId('65aa8876d792947f8dbaf957'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
}
{
```

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

```
>_MONGOSH
> db.products.find({}, {product_name: 1 , product_material: 1})
< {
  _id: ObjectId('65aa8876d792947f8dbaf956'),
  product_name: 'Intelligent Fresh Chips',
  product_material: 'Concrete'
}
{
  _id: ObjectId('65aa8876d792947f8dbaf95e'),
  product_name: 'Awesome Wooden Ball',
  product_material: 'Soft'
}
{
  _id: ObjectId('65aa8876d792947f8dbaf960'),
  product_name: 'Unbranded Wooden Cheese',
  product_material: 'Soft'
}
{
  _id: ObjectId('65aa8876d792947f8dbaf964'),
  product_name: 'Ergonomic Cotton Hat',
  product_material: 'Rubber'
}
{
  _id: ObjectId('65aa8876d792947f8dbaf965'),
  product_name: 'Small Soft Chair',
  product_material: 'Cotton'
}
```

6. Find the product with a row id of 10

```
> db.products.findOne({id:'10'})
< {
  _id: ObjectId('65aa8876d792947f8dbaf95f'),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 84,
  product_material: 'Frozen',
  product_color: 'indigo'
}
Atlas atlas-6bcx3y-shard-0 [primary] guvi>
```

7. Find only the product name and product material

```
>_MONGOSH
> db.products.find({}, {_id: 0, product_name: 1 , product_material: 1}).toArray()
< [
  {
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  },
  { product_name: 'Awesome Wooden Ball', product_material: 'Soft' },
  { product_name: 'Unbranded Wooden Cheese', product_material: 'Soft' },
  { product_name: 'Ergonomic Cotton Hat', product_material: 'Rubber' },
  { product_name: 'Small Soft Chair', product_material: 'Cotton' },
  { product_name: 'Licensed Steel Car', product_material: 'Cotton' },
  { product_name: 'Incredible Metal Car', product_material: 'Fresh' },
  { product_name: 'Generic Wooden Pizza', product_material: 'Frozen' },
  r
```

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

```
>_MONGOSH
> db.products.find({}, {product_material: 'Soft'}).toArray()
< [
  {
    _id: ObjectId('65aa8876d792947f8dbaf956'),
    product_material: 'Soft'
  },
  {
    _id: ObjectId('65aa8876d792947f8dbaf95e'),
    product_material: 'Soft'
  },
]
```

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

```
> db.products.find({$and: [{product_color: 'indigo'}, {product_price: 492}]}).toArray()
< []
Atlas atlas-6bcx3y-shard-0 [primary] guvi>
```

10. Delete the products which product price value are same

```
Atlas atlas-6bcx3y-shard-0 [primary] guvi> db.products.aggregate([
  {
    $group: {
      _id: { product_price: "$product_price" },
      duplicates: { $addToSet: "$_id" },
      count: { $sum: 1 }
    }
  },
  {
    $match: {
      count: { $gt: 1 }
    }
  }
]).forEach(function(doc) {
  doc.duplicates.shift(); // Keep one document, remove others
  db.products.deleteMany({ _id: { $in: doc.duplicates } });
});
```