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?

```
> db.products.find().pretty()
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e3"),
        "id" : "1",
        "product_name" : "Intelligent Fresh Chips",
        "product_price" : 655,
        "product_material" : "Concrete",
        "product_color" : "mint green"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e4"),
        "id" : "2",
        "product_name" : "Practical Fresh Sausages",
        "product_price" : 911,
        "product_material" : "Cotton",
        "product_color" : "indigo"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e5"),
        "id" : "3",
        "product_name" : "Refined Steel Car",
        "product_price" : 690,
        "product_material" : "Rubber",
        "product_color" : "gold"
}
```

Solution: **db.products.find().pretty()**

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

```
> db.products.find({product_price:{$gt:400,$lt:800}}).pretty()
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e3"),
        "id" : "1",
        "product_name" : "Intelligent Fresh Chips",
        "product_price" : 655,
        "product_material" : "Concrete",
        "product_color" : "mint green"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e5"),
        "id" : "3",
        "product_name" : "Refined Steel Car",
        "product_price" : 690,
        "product_material" : "Rubber",
        "product_color" : "gold"
}
```

Solution: db.products.find({product_price:{$gt:400,$lt:800}}).pretty()

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

```
> db.products.find({product_price:{$not:{$gte:400,$lte:600}}}).pretty()
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e3"),
        "id" : "1",
        "product_name" : "Intelligent Fresh Chips",
        "product_price" : 655,
        "product_material" : "Concrete",
        "product_color" : "mint green"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e4"),
        "id" : "2",
        "product_name" : "Practical Fresh Sausages",
        "product_price" : 911,
        "product_material" : "Cotton",
        "product_color" : "indigo"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e5"),
        "id" : "3",
```

Solution: db.products.find({product_price:{$not:{$gte:400,$lte:600}}}).pretty()

4) **List the four product which are grater than 500 in price?**

```
> db.products.find({product_price:{$gt:500}}).limit(4).pretty()
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e3"),
        "id" : "1",
        "product_name" : "Intelligent Fresh Chips",
        "product_price" : 655,
        "product_material" : "Concrete",
        "product_color" : "mint green"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e4"),
        "id" : "2",
        "product_name" : "Practical Fresh Sausages",
        "product_price" : 911,
        "product_material" : "Cotton",
        "product_color" : "indigo"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e5"),
        "id" : "3",
        "product_name" : "Refined Steel Car",
        "product_price" : 690,
        "product_material" : "Rubber",
        "product_color" : "gold"
```

Solution: db.products.find({product_price:{$gt:500}}).limit(4).pretty()

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

```
@(shell):1:3
> db.products.find({},{product_name:1,product_material:1}).pretty()
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e3"),
        "product_name" : "Intelligent Fresh Chips",
        "product_material" : "Concrete"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e4"),
        "product_name" : "Practical Fresh Sausages",
        "product_material" : "Cotton"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e5"),
        "product_name" : "Refined Steel Car",
        "product_material" : "Rubber"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e6"),
        "product_name" : "Gorgeous Plastic Pants",
        "product_material" : "Soft"
}
```

Solution: db.products.find({},{product_name:1,product_material:1}).pretty()

6) **Find the product with a row id of 10?**

```
> db.products.find({id:"10"}).pretty()
{
        "_id" : ObjectId("65df53bf45806e8457f0e8ec"),
        "id" : "10",
        "product_name" : "Generic Wooden Pizza",
        "product_price" : 84,
        "product_material" : "Frozen",
        "product_color" : "indigo"
}
>
```

Solution: db.products.find({id:"10"}).pretty()

7) **Find only the product name and product material?**

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

Solution: db.products.find({},{product_name:1,product_material:1,id:0}).pretty()

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

```
> db.products.find({product_material: 'Soft'}).pretty()
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e6"),
        "id" : "4",
        "product_name" : "Gorgeous Plastic Pants",
        "product_price" : 492,
        "product_material" : "Soft",
        "product_color" : "plum"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8eb"),
        "id" : "9",
        "product_name" : "Awesome Wooden Ball",
        "product_price" : 28,
        "product_material" : "Soft",
        "product_color" : "azure"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8ed"),
        "id" : "11",
```

Solution: db.products.find({product_material: 'Soft'}).pretty()

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

```
> db.products.find({$or:[{product_color:'indigo'},{product__price:'492.00'}]}).pretty()
{
        "_id" : ObjectId("65df53bf45806e8457f0e8e4"),
        "id" : "2",
        "product_name" : "Practical Fresh Sausages",
        "product_price" : 911,
        "product_material" : "Cotton",
        "product_color" : "indigo"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8ec"),
        "id" : "10",
        "product_name" : "Generic Wooden Pizza",
        "product_price" : 84,
        "product_material" : "Frozen",
        "product_color" : "indigo"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8f3"),
        "id" : "17",
        "product_name" : "Incredible Metal Car",
        "product_price" : 36,
        "product_material" : "Fresh",
        "product_color" : "indigo"
}
{
        "_id" : ObjectId("65df53bf45806e8457f0e8fb"),
        "id" : "25",
        "product_name" : "Licensed Steel Car",
        "product_price" : 20,
        "product_material" : "Cotton",
        "product_color" : "indigo"
}
```

Solution:

db.products.find({$or:[{product_color:'indigo'},{product__price:'492.00'}]}).pretty()

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

```
> db.products.aggregate([
...   {
...     $group: {
...       _id: "$product_price",
...       count: { $sum: 1 },
...       ids: { $push: "$_id" }
...     }
...   },
...   {
...     $match: {
...       count: { $gt: 1 }
...     }
...   }
... ]).forEach(function(doc) {
...   doc.ids.shift();
...   db.products.deleteMany({_id: {$in: doc.ids}});
... });
> print("Documents with duplicate product prices have been deleted successfully.");
Documents with duplicate product prices have been deleted successfully.
>
```

Solution :

```
db.products.aggregate([
 {
  $group: {
   _id: "$product_price",
count: { $sum: 1 },
    ids: { $push: "$_id" }
  }
 },
 {
  $match: {
count: { $gt: 1 }
  }
 }
]).forEach(function(doc) {
 doc.ids.shift();
 db.products.deleteMany({_id: {$in: doc.ids}});
});
print("Documents with duplicate product prices have been deleted successfully.");
```