

MongoDB Day 1

For the following question write the corresponding MongoDB queries

Inserted json data:

dbtask> db.products.insertMany([inserted json data])

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('660a393c071a7dd49c9f9923'),
    '1': ObjectId('660a393c071a7dd49c9f9924'),
    '2': ObjectId('660a393c071a7dd49c9f9925'),
    '3': ObjectId('660a393c071a7dd49c9f9926'),
    '4': ObjectId('660a393c071a7dd49c9f9927'),
    '5': ObjectId('660a393c071a7dd49c9f9928'),
    '6': ObjectId('660a393c071a7dd49c9f9929'),
    '7': ObjectId('660a393c071a7dd49c9f992a'),
    '8': ObjectId('660a393c071a7dd49c9f992b'),
    '9': ObjectId('660a393c071a7dd49c9f992c'),
    '10': ObjectId('660a393c071a7dd49c9f992d'),
    '11': ObjectId('660a393c071a7dd49c9f992e'),
    '12': ObjectId('660a393c071a7dd49c9f992f'),
    '13': ObjectId('660a393c071a7dd49c9f9930'),
    '14': ObjectId('660a393c071a7dd49c9f9931'),
    '15': ObjectId('660a393c071a7dd49c9f9932'),
    '16': ObjectId('660a393c071a7dd49c9f9933'),
    '17': ObjectId('660a393c071a7dd49c9f9934'),
    '18': ObjectId('660a393c071a7dd49c9f9935'),
    '19': ObjectId('660a393c071a7dd49c9f9936'),
    '20': ObjectId('660a393c071a7dd49c9f9937'),
    '21': ObjectId('660a393c071a7dd49c9f9938'),
    '22': ObjectId('660a393c071a7dd49c9f9939'),
    '23': ObjectId('660a393c071a7dd49c9f993a'),
    '24': ObjectId('660a393c071a7dd49c9f993b')
  }
}
```

1. Find all the information about each products

db.products.find().pretty()

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

```
db.products.find(  
  {  
    product_price:{$gte:400,$lte:800}  
  },  
  {"product_price":1,"_id":0}  
)
```

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

```
db.products.find(  
  {$nor: [  
    {product_price:{$gte:400,$lte:600}}  
  ]},  
  {product_price:1,_id:0} )
```

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

```
db.products.aggregate([  
  {$sort:{product_price:-1}},{$limit:4}  
)
```

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

```
db.products.find({},  
  {product_name:1,product_material:1,_id:0}  
)
```

6. Find the product with a row id of 10

Using aggregate :

```
db.products.aggregate([
```

```
{ $match: { id: "10" } }  
])
```

Using find :

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

7. Find only the product name and product material

Using aggregate:

```
db.products.aggregate([  
  { $project: { product_name: 1, product_material: 1, _id: 0 } }  
])
```

:

```
db.products.find({}, { product_material: 1,  
  product_name: 1, _id: 0 })
```

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

```
db.products.aggregate([  
  { $match: { product_material: "Soft" } }  
])
```

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

```
db.products.aggregate([  
  { $project: { product_color: "indigo", product_price: "492" } }  
])
```

10. Delete the products which product price value are same

```
db.products.aggregate([
  {
    $group: { _id: "$product_price", count: { $sum: 1 }}
  },
  {
    $match: {count: { $gt: 1 }}
  }
]).forEach(function(doc) {
  db.products.deleteMany({ product_price: doc._id });
});
```