

# Database- Day -3: MongoDB Task-1

1.Find all the information about each products

```
>_MONGOSH
> db.Task3.find();
< {
  _id: ObjectId('6691026e4381e4277b20ad05'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('6691026e4381e4277b20ad06'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
}
```

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

```
> db.Task3.find({
  product_price: { $gte: 400, $lte: 900 }
});
< {
  _id: ObjectId('6691026e4381e4277b20ad05'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('6691026e4381e4277b20ad07'),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
  product_material: 'Rubber',
  product_color: 'gold'
}
```

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

```

> db.Task3.find({
  $or: [
    { product_price: { $gte: 400, $lte: 900 } }
  ]
});
< {
  _id: ObjectId('6691026e4381e4277b20ad05'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('6691026e4381e4277b20ad07'),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
  product_material: 'Rubber',
  product_color: 'gold'
}

```

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

```

> db.Task3.find( {product_price:{$gte:500}}
).limit(4);
< {
  _id: ObjectId('6691026e4381e4277b20ad05'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('6691026e4381e4277b20ad06'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
}

```

```

db.Task3.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'
}

```

Find the product name and product material of each products

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

```

> db.Task3.find({id:'10'});
< {
  _id: ObjectId('6691026e4381e4277b20ad0e'),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 84,
  product_material: 'Frozen',
  product_color: 'indigo'
}
Task3> |

```

## 7. Find only the product name and product material

```
> db.Task3.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'
}
```

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

```
> db.Task3.find({ product_material: "Soft" });
< {
  _id: ObjectId('6691026e4381e4277b20ad08'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
}
{
  _id: ObjectId('6691026e4381e4277b20ad0d'),
  id: '9',
  product_name: 'Awesome Wooden Ball',
  product_price: 28,
  product_material: 'Soft',
  product_color: 'azure'
}
{
  _id: ObjectId('6691026e4381e4277b20ad0f'),
  id: '11',
  product_name: 'Unbranded Wooden Cheese',
  product price: 26.
```

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

```
Task3> db.Task3.find({  
  $and: [  
    { product_color: "indigo" },  
    { product_price: 492.00 }  
  ]  
});
```

10.Delete the products which product price value are 28

```
> db.Task3.deleteMany({ product_price: 28 });
```

```
< {
```

```
  acknowledged: true,
```

```
  deletedCount: 1
```

```
}
```

```
Task3>
```