

MongoDB Day 1

1. Find all the information about each products.
`db.taskproduct.find()`

```
> db.taskproduct.find();
{
  "_id": ObjectId("65e3ecc723aec65bbed71ceb"),
  "id": "1",
  "product_name": "Intelligent Fresh Chips",
  "product_price": 655,
  "product_material": "Concrete",
  "product_color": "mint green"
}
{
  "_id": ObjectId("65e3ecc723aec65bbed71cec"),
  "id": "2",
  "product_name": "Practical Fresh Sausages",
  "product_price": 911,
  "product_material": "Cotton",
  "product_color": "indigo"
}
{
  "_id": ObjectId("65e3ecc723aec65bbed71ced"),
  "id": "3",
  "product_name": "Refined Steel Car",

```

2. Find the product price which are between 400 to 800
`db.taskproduct.find`
{
 "product_price": {
 \$gte: 400, \$lte: 800
 }
}
)

```
> db.taskproduct.find({
  "product_price": {
    $gte: 400,
    $lte: 800
  }
})
{
  "_id": ObjectId("65e3ecc723aec65bbed71ceb"),
  "id": "1",
  "product_name": "Intelligent Fresh Chips",
  "product_price": 655,
  "product_material": "Concrete",
  "product_color": "mint green"
}
{
  "_id": ObjectId("65e3ecc723aec65bbed71ced"),
  "id": "3",

```

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

```
}
> db.taskproduct.find({ "product_price": {
    $lt: 400,
    $gt: 600
  }
});
<
mongodbtask>
```

- List the four product which are greater than 500 in price

```
Db.taskproduct.find({"product_price":{
$gt:500}}).limit(4)
db.taskproduct.find({
  "product_price": {
    $gt: 500
  }
}).limit(4)
{
  _id: ObjectId('65e3ecc723aec65bbed71ceb'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('65e3ecc723aec65bbed71cec'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
}
{
  _id: ObjectId('65e3ecc723aec65bbed71ced'),
  id: '3',
```

- Find the product name and product material of each products

```
Db.taskproduct.find({}, {
  "_id": 0,
  "product_name": 1,
  "product_material": 1
}))
```

```
db.taskproduct.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'
}
```

6. Find the product with a row id of 10

```
Db.taskproduct.findone({
  "id": "10"
})
```

```

> db.taskproduct.findOne({
  "id": "10"
})
< {
  _id: ObjectId('65e3ecc723aec65bbed71cf4'),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 84,
  product_material: 'Frozen',
  product_color: 'indigo'
}
mongodbtask> |

```

7. Find only the product name and product material

```

Db.taskkproduct.find({}, {
  "_id": 0,
  "product_name": 1,
  "product_material": 1})

```

```

> db.taskproduct.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',

```

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

```

Db.taskproduct.find({
  "product_material": {

```

```
$regex:/soft/i}
```

```
}}
```

```
> db.taskproduct.find({
  "product_material": {
    $regex: /soft/i
  }
})
< {
  _id: ObjectId('65e3ecc723aec65bbbed71cee'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
}
{
  _id: ObjectId('65e3ecc723aec65bbbed71cf3'),
  id: '9',
  product_name: 'Awesome Wooden Ball',
  product_price: 28,
  product_material: 'Soft',
  product_color: 'azure'
}
{
  _id: ObjectId('65e3ecc723aec65bbbed71cf5'),
  id: '11',
  product_name: 'Unbranded Wooden Cheese',
  product_price: 26,
```

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

10. Delete the products which product price value are same