MongoDB Task

1.Find all the information about each products

## Answer: db.products.find();

few example product for this answer:

{  
 \_id: ObjectId(‘65e02fdde3661802d1df5b60’),  
 id: ‘1’,  
 product\_name: ‘Intelligent Fresh Chips’,  
 product\_price: 655,  
 product\_material: ‘Concrete’,  
 product\_color: ‘mint green’  
}

{  
 \_id: ObjectId(‘65e02fdde3661802d1df5b61’),  
 id: ‘2’,  
 product\_name: ‘Practical Fresh Sausages’,  
 product\_price: 911,  
 product\_material: ‘Cotton’,  
 product\_color: ‘indigo’  
}

{

\_id: ObjectId('65e02fdde3661802d1df5b62'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b63'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b64'),

id: '5',

product\_name: 'Sleek Cotton Chair',

product\_price: 33,

product\_material: 'Fresh',

product\_color: 'black'

}…..etc show 25 products

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

## Answer: db.products.find({product\_price:{$gt:400,$lt:800}});

few example product for this answer:

{

\_id: ObjectId('65e02fdde3661802d1df5b60'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b62'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b63'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b65'),

id: '6',

product\_name: 'Awesome Wooden Towels',

product\_price: 474,

product\_material: 'Plastic',

product\_color: 'orange'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b66'),

id: '7',

product\_name: 'Practical Soft Shoes',

product\_price: 500,

product\_material: 'Rubber',

product\_color: 'pink'

}

{

\_id: ObjectId('65e030a8e3661802d1df5b79'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

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

# ans: db.products.find({product\_price:{$not :{$gt:400,$lt:800}}});

few example product for this answer:

{

\_id: ObjectId('65e02fdde3661802d1df5b61'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b67'),

id: '8',

product\_name: 'Incredible Steel Hat',

product\_price: 78,

product\_material: 'Rubber',

product\_color: 'violet'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b69'),

id: '10',

product\_name: 'Generic Wooden Pizza',

product\_price: 84,

product\_material: 'Frozen',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b6b'),

id: '12',

product\_name: 'Unbranded Plastic Salad',

product\_price: 89,

product\_material: 'Wooden',

product\_color: 'pink'

}

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

## Answer: db.products.find({product\_price:{$gt:500}}).limit(4);

{

\_id: ObjectId('65e02fdde3661802d1df5b60'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b61'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e02fdde3661802d1df5b62'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

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

## Answer: db.products.find({},{'product\_name':1,'product\_material':1});

few example product for this answer:

{

\_id: ObjectId('65e04a894a16f7ceeed5662c'),

product\_name: 'Intelligent Fresh Chips',

product\_material: 'Concrete'

}

{

\_id: ObjectId('65e04a894a16f7ceeed5662d'),

product\_name: 'Practical Fresh Sausages',

product\_material: 'Cotton'

}

{

\_id: ObjectId('65e04a894a16f7ceeed5662e'),

product\_name: 'Refined Steel Car',

product\_material: 'Rubber'

}

{

\_id: ObjectId('65e04a894a16f7ceeed5662f'),

product\_name: 'Gorgeous Plastic Pants',

product\_material: 'Soft'

}

6. Find the product with a row id of 10

## Answer: db.products.findOne({id:"10"});

{

\_id: ObjectId('65e04a894a16f7ceeed56635'),

id: '10',

product\_name: 'Generic Wooden Pizza',

product\_price: 84,

product\_material: 'Frozen',

product\_color: 'indigo'

}….

7. Find only the product name and product material

## Answer: db.products.find({},{'\_id':0,'product\_name':1,'product\_material':1});

few example product for this answer:

{

product\_name: 'Intelligent Fresh Chips',

product\_material: 'Concrete'

}

{

product\_name: 'Practical Fresh Sausages',

product\_material: 'Cotton'

}

{

product\_name: 'Refined Steel Car',

product\_material: 'Rubber'

}….

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

## Answer: db.products.find({product\_material:'Soft'});

{

\_id: ObjectId('65e04a894a16f7ceeed5662f'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

}

{

\_id: ObjectId('65e04a894a16f7ceeed56634'),

id: '9',

product\_name: 'Awesome Wooden Ball',

product\_price: 28,

product\_material: 'Soft',

product\_color: 'azure'

}

{

\_id: ObjectId('65e04a894a16f7ceeed56636'),

id: '11',

product\_name: 'Unbranded Wooden Cheese',

product\_price: 26,

product\_material: 'Soft',

product\_color: 'black'

}

{

"id": "19",

"product\_name": "Intelligent Cotton Chips",

"product\_price": 46.00,

"product\_material": "Soft",

"product\_color": "azure"

}

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

## Answer:db.products.find({$or:[{product\_color:'indigo'},{product\_price:492.00});

few example product for this answer:

{

\_id: ObjectId('65e04a894a16f7ceeed5662d'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e04a894a16f7ceeed5662f'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

}

{

\_id: ObjectId('65e04a894a16f7ceeed56635'),

id: '10',

product\_name: 'Generic Wooden Pizza',

product\_price: 84,

product\_material: 'Frozen',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e04a894a16f7ceeed5663c'),

id: '17',

product\_name: 'Incredible Metal Car',

product\_price: 36,

product\_material: 'Fresh',

product\_color: 'indigo'

}

10. Delete the products which product price value are same

## Answer: db.products.aggregate([

## { $group: { \_id: "$product\_price", count: { $sum: 1 } } },

## { $match: { count: { $gt: 1 } } }

## ]).forEach((e) => {

## db.products.deleteMany({ product\_price: e.\_id });

## });

