

Mondo db queries- task1

0. Find all the information about each products

```
}
mongodbquery> db.myproducts.find();
[
  { _id: ObjectId('65e1bc847ce6572c9ff31938'), id: '1' },
  { _id: ObjectId('65e1bf4d596d50cc5774cfef'), id: '1' },
  {
    _id: ObjectId('65e1bfe7596d50cc5774cff0'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  { _id: ObjectId('65e20270596d50cc5774cff1'), id: '1' },
  { _id: ObjectId('65e202f5596d50cc5774cff2'), id: '2' },
  {
    _id: ObjectId('65e203db596d50cc5774cff3'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('65e203db596d50cc5774cff4'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {
    _id: ObjectId('65e203db596d50cc5774cff5'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
  .....
]
```

0. Find the product price which are between 400 to 800
mongodbquery> db.myproducts.find({ "product_price": { \$gte: 400, \$lte: 800 } });

```
[
  {
    _id: ObjectId('65e22d2e596d50cc5774d024'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('65e22d2e596d50cc5774d026'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {
    _id: ObjectId('65e22d2e596d50cc5774d027'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
]
```

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

mongodbquery> db.myproducts.find({'product_price':{\$not:{\$gte: 400,\$lte:600}}});

```
[
  {
    _id: ObjectId('65e22d2e596d50cc5774d024'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',

```



```

    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  }
]

```

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

```

mongodbquery> db.myproducts.find({}, { "product_name": 1,
"product_material": 1});
[
  {
    _id: ObjectId('65e22d2e596d50cc5774d024'),
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  },
  {
    _id: ObjectId('65e22d2e596d50cc5774d025'),
    product_name: 'Practical Fresh Sausages',
    product_material: 'Cotton'
  },
  {
    _id: ObjectId('65e22d2e596d50cc5774d026'),
    product_name: 'Refined Steel Car',
    product_material: 'Rubber'
  },
  {
    _id: ObjectId('65e22d2e596d50cc5774d027'),
    product_name: 'Gorgeous Plastic Pants',
    product_material: 'Soft'
  },
  .....
]

```

6 . Find the product with a row id of 10

```

mongodbquery> db.myproducts.find({'id': '10'});
[
  {
    _id: ObjectId('65e20572596d50cc5774cffb'),
    id: '10',
    product_name: 'Generic Wooden Pizza',
    product_price: 84,
    product_material: 'Frozen',
    product_color: 'indigo'
  }
]

```

]

.....

0. Find only the product name and product material

```
mongodbquery> db.myproducts.find({}, { "product_name": 1,
"product_material": 1, "_id": 0 });
[
  {},
  {},
  {
    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' },
  {
    product_name: 'Awesome Wooden Towels',
    product_material: 'Plastic'
  },
  { product_name: 'Practical Soft Shoes', product_material: 'Rubber' },
  { product_name: 'Incredible Steel Hat', product_material: 'Rubber' },
  { product_name: 'Awesome Wooden Ball', product_material: 'Soft' },
  { product_name: 'Generic Wooden Pizza', product_material: 'Frozen' },
  { product_name: 'Unbranded Wooden Cheese', product_material: 'Soft' },
  {
    product_name: 'Unbranded Plastic Salad',
    product_material: 'Wooden'
  },
]
```

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

```

mongodbquery> db.myproducts.find({'product_material':{'$regex:/soft/i'}});
[
  {
    _id: ObjectId('65e203db596d50cc5774cff5'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
  {
    _id: ObjectId('65e20572596d50cc5774cffa'),
    id: '9',
    product_name: 'Awesome Wooden Ball',
    product_price: 28,
    product_material: 'Soft',
    product_color: 'azure'
  },
  {
    _id: ObjectId('65e20572596d50cc5774cffc'),
    id: '11',
    product_name: 'Unbranded Wooden Cheese',
    product_price: 26,
    product_material: 'Soft',
    product_color: 'black'
  },
]

```

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

Type "it" for more

```

mongodbquery> db.myproducts.find({'$and':[{'product_color':'indigo'},
{'product_price':492}]});

```

```

mongodbquery>

```

note: AND operator works when both condition are true. So, there is no values having product_color as "indigo"

And product_price as 492 in same product. Here, no document ratifying both conditions.

10 .Delete the products which product price value are same

```

db.myproducts.aggregate([

```

```
    { $group: { '_id': '$product_price', count: { $sum: 1 } } },  
    { $match: { count: { $gt: 1 } } }  
  ]).forEach((e) => {  
    db.myproducts.deleteMany({ product_price: e._id });  
  });
```