**MONGODB - TASK - 1**

1. **Find all the information about each products**

QUERY: db.products.find();

Output:

{

\_id: ObjectId('65e056c674c495678e80a9ef'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}{

\_id: ObjectId('65e056c674c495678e80a9ef'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

…

1. **Find the product price which are between 400 to 800**

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

Output: {

\_id: ObjectId('65e056c674c495678e80a9ee'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

{

\_id: ObjectId('65e056c674c495678e80a9f0'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}…

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

QUERY: db.products.find({ $nor: [ { product\_price: { $gte: 400, $lte: 600 } } ] })

OUTPUT: {

\_id: ObjectId('65e056c674c495678e80a9ef'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e056c674c495678e80a9ee'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}…

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

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

OUTPUT: {

\_id: ObjectId('65e056c674c495678e80a9ef'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

{

\_id: ObjectId('65e056c674c495678e80a9f0'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

{

\_id: ObjectId('65e056c674c495678e80a9ee'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}

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

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

OUTPUT: {

\_id: ObjectId('65e056c674c495678e80a9f0'),

product\_name: 'Refined Steel Car',

product\_material: 'Rubber'

}

{

\_id: ObjectId('65e056c674c495678e80a9ef'),

product\_name: 'Practical Fresh Sausages',

product\_material: 'Cotton'

}

{

\_id: ObjectId('65e056c674c495678e80a9ee'),

product\_name: 'Intelligent Fresh Chips',

product\_material: 'Concrete'

}

**6. Find the product with a row id of 10**

QUERY: db.products.find({'id':'10'});

OUTPUT: {

\_id: ObjectId('65e056c674c495678e80a9f7'),

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**

QUERY:db.products.find({},{'product\_name':1,'product\_material':1,

'\_id':0});

OUTPUT: {

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'

}

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

QUERY: db.products.find({'product\_material':{$eq:'Soft'}});

OUTPUT: {

\_id: ObjectId('65e056c674c495678e80a9f8'),

id: '11',

product\_name: 'Unbranded Wooden Cheese',

product\_price: 26,

product\_material: 'Soft',

product\_color: 'black'

}

{

\_id: ObjectId('65e056c674c495678e80a9f6'),

id: '9',

product\_name: 'Awesome Wooden Ball',

product\_price: 28,

product\_material: 'Soft',

product\_color: 'azure'

}

{

\_id: ObjectId('65e056c674c495678e80a9f1'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum' }

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

QUERY:db.products.find({$and:[{'product\_color':'indigo'},{'product\_price':492}]});

OUTPUT: There is no documents with product\_color:”indigo” and product\_price:492

1. **Delete the products which product price value are same**

QUERY:db.products.aggregate([

{

$group: {

\_id: "$product\_price",

count: { $sum: 1 },

ids: { $push: "$\_id" }

}

},

{

$match: {

count: { $gt: 1 }

}

},

{

$unwind: "$ids"

},

{

$project: {

\_id: 0,

id: "$ids"

}

}

]).forEach(function(doc) {

db.products.deleteOne({ "\_id": doc.id });

});

OUTPUT: DeleteResult { acknowledged: true, deletedCount: 2 }