1. Find all the information about each products?

Query = db.products.find()

Response = {

\_id: ObjectId('65e02f24bd9d8c3a9c718375'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

},{

\_id: ObjectId('65e02f24bd9d8c3a9c718376'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}

.

.

.

.

{

\_id: ObjectId('65e02f24bd9d8c3a9c718388'),

id: '20',

product\_name: 'Handcrafted Wooden Bacon',

product\_price: 36,

product\_material: 'Concrete',

product\_color: 'lime'

}

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

Query: db.products.find({‘products\_\_price’: {$gt:400, $lt:800}})

{

\_id: ObjectId('65e02f24bd9d8c3a9c718375'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}……

{

\_id: ObjectId('65e02f24bd9d8c3a9c71837b'),

id: '7',

product\_name: 'Practical Soft Shoes',

product\_price: 500,

product\_material: 'Rubber',

product\_color: 'pink'

}

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

Query = db.products.find({'product\_price':{$not: {$gt:400, $lt:600}}});

Response: {

\_id: ObjectId('65e02f24bd9d8c3a9c718375'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

},

…..{

{

\_id: ObjectId('65e02f24bd9d8c3a9c71838b'),

id: '23',

product\_name: 'Intelligent Steel Pizza',

product\_price: 95,

product\_material: 'Cotton',

product\_color: 'azure'

}

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

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

Response: {

\_id: ObjectId('65e02f24bd9d8c3a9c718375'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

}, {

\_id: ObjectId('65e02f24bd9d8c3a9c718376'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}, {

\_id: ObjectId('65e02f24bd9d8c3a9c718377'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

}

There are only 3 products whose values are more than 500. But If we include 500 also, Then we get 4 products. For that query is as follows:

db.products.find({'product\_price': {$gte:500}});

Response : The 4th object{

\_id: ObjectId('65e02f24bd9d8c3a9c71837b'),

id: '7',

product\_name: 'Practical Soft Shoes',

product\_price: 500,

product\_material: 'Rubber',

product\_color: 'pink'

}

1. Find the product name and product material of each products?

Query: db.products.find({},{'\_id':0, 'product\_price' : 1,'product\_name' : 1});

Response : {

product\_name: 'Intelligent Fresh Chips',

product\_price: 655

},

…..

{

product\_name: 'Handcrafted Wooden Bacon',

product\_price: 36

}

1. Find the product with a row id of 10?

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

Response: {

\_id: ObjectId('65e02f24bd9d8c3a9c71837e'),

id: '10',

product\_name: 'Generic Wooden Pizza',

product\_price: 84,

product\_material: 'Frozen',

product\_color: 'indigo'

}

1. Find only the product name and product material?

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

Response: {

product\_name: 'Intelligent Fresh Chips',

product\_material: 'Concrete'

}…. {

product\_name: 'Handcrafted Wooden Bacon',

product\_material: 'Concrete'

}

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

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

Response: {

\_id: ObjectId('65e02f24bd9d8c3a9c718378'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

}…. {

\_id: ObjectId('65e02f24bd9d8c3a9c718387'),

id: '19',

product\_name: 'Intelligent Cotton Chips',

product\_price: 46,

product\_material: 'Soft',

product\_color: 'azure'

}

1. Find products which contain product color indigo  and product price 492.00?

Query: db.products.find({$or:[{'product\_price' : {$eq : 592}},{'product\_color': {$eq:'indigo'}}]});

Response: {

\_id: ObjectId('65e02f24bd9d8c3a9c718376'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}… {

\_id: ObjectId('65e02f24bd9d8c3a9c718378'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

}

1. Delete the products which product price value are same