**Mongo DB Task -1**

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

Answer:db.products.find()

Output:

{

\_id: ObjectId('65e03aa81a30a72d0e3b3522'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3523'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

}……….etc

1. **Find the Product Price which are between 400 and 800**.

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

Output:

{

\_id: ObjectId('65e03aa81a30a72d0e3b3522'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3524'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3525'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3527'),

id: '6',

product\_name: 'Awesome Wooden Towels',

product\_price: 474,

product\_material: 'Plastic',

product\_color: 'orange'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3528'),

id: '7',

product\_name: 'Practical Soft Shoes',

product\_price: 500,

product\_material: 'Rubber',

product\_color: 'pink'

}

**3)Find the Product Price which are not between 400 and 800.**

Answer: db.products.find({“product\_price”:{$not:{$gt:400,$lt:800}}})

Output:

{

\_id: ObjectId('65e03aa81a30a72d0e3b3523'),

id: '2',

product\_name: 'Practical Fresh Sausages',

product\_price: 911,

product\_material: 'Cotton',

product\_color: 'indigo'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3526'),

id: '5',

product\_name: 'Sleek Cotton Chair',

product\_price: 33,

product\_material: 'Fresh',

product\_color: 'black'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3529'),

id: '8',

product\_name: 'Incredible Steel Hat',

product\_price: 78,

product\_material: 'Rubber',

product\_color: 'violet'

}……….etc

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

Answer: db.products.find({“product\_price”:{$gt:400,$lt:800}}).limit(4)

Output:

{

\_id: ObjectId('65e03aa81a30a72d0e3b3522'),

id: '1',

product\_name: 'Intelligent Fresh Chips',

product\_price: 655,

product\_material: 'Concrete',

product\_color: 'mint green'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3524'),

id: '3',

product\_name: 'Refined Steel Car',

product\_price: 690,

product\_material: 'Rubber',

product\_color: 'gold'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3525'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3527'),

id: '6',

product\_name: 'Awesome Wooden Towels',

product\_price: 474,

product\_material: 'Plastic',

product\_color: 'orange'

}

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

Anwer: db.products.find({},{ product\_name: 1, product\_material: 1})

Output:

{

\_id: ObjectId('65e03aa81a30a72d0e3b3522'),

product\_name: 'Intelligent Fresh Chips',

product\_material: 'Concrete'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3523'),

product\_name: 'Practical Fresh Sausages',

product\_material: 'Cotton'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3524'),

product\_name: 'Refined Steel Car',

product\_material: 'Rubber'

}…………etc

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

Answer: db.products.find({id: "10"})

Output:

{

\_id: ObjectId('65e03aa81a30a72d0e3b352b'),

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({},{ 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'}………etc

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

Answer:db.products.find({product\_material: { "$eq": "Soft" }})

Output:

{

\_id: ObjectId('65e03aa81a30a72d0e3b3525'),

id: '4',

product\_name: 'Gorgeous Plastic Pants',

product\_price: 492,

product\_material: 'Soft',

product\_color: 'plum'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b352a'),

id: '9',

product\_name: 'Awesome Wooden Ball',

product\_price: 28,

product\_material: 'Soft',

product\_color: 'azure'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b352c'),

id: '11',

product\_name: 'Unbranded Wooden Cheese',

product\_price: 26,

product\_material: 'Soft',

product\_color: 'black'

},

{

\_id: ObjectId('65e03aa81a30a72d0e3b3534'),

id: '19',

product\_name: 'Intelligent Cotton Chips',

product\_price: 46,

product\_material: 'Soft',

product\_color: 'azure'

}

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

Answer:db.products.find({ "$and": [{product\_color: "indigo" },{product\_price: 492.00 }]})

Output :

No documents are there with product\_color:”indigo” and product\_price:492

**10)Delete the products which product price value are same.**

Answer:db.products.aggregate([ { "$group": { "\_id": { product\_price: "$product\_price" }, total: { $sum: 1 } } }, { $match: { total: { $gt: 1 } } } ]).map(val=> db.collection.deleteMany({"product\_price": val.\_id.product\_price}))

Output:

DeleteResult { acknowledged: true, deletedCount: 2 }