

MongoDB Day 1

Product JSON: <https://github.com/rvsp/database/blob/master/mongodb/product.json>

MongoDB Queries

1. Find all the information about each Products.

ANS: `db.shop.find();`

```
db.shop.find();
{
  _id: ObjectId('666c325a1cca5014bf4d9e55'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('666c325a1cca5014bf4d9e56'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
}
```

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

ANS: `db.shop.find({product_price : {$gt:400 , $lt:800}});`

```

db.shop.find({product_price:{$gt:400 , $lt:800}});
{
  _id: ObjectId('666c325a1cca5014bf4d9e55'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('666c325a1cca5014bf4d9e57'),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
  product_material: 'Rubber',

```

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

ANS: db.shop.find({product_price :{\$not: {\$gt:400 , \$lt:800}}});

```

db.shop.find({product_price:{$not :{$gt:400 , $lt:800}}});
{
  _id: ObjectId('666c325a1cca5014bf4d9e56'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
}
{
  _id: ObjectId('666c325a1cca5014bf4d9e59'),
  id: '5',
  product_name: 'Sleek Cotton Chair',
  product_price: 33,
  product_material: 'Fresh',
  product_color: 'black'
}

```

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

ANS: db.shop.find({product_price{\$gt:500}}).limit(4);

```

Type "it" for more
> db.shop.find({ product_price:{$gt:500}}).limit(4);
< {
  _id: ObjectId('666c325a1cca5014bf4d9e55'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('666c325a1cca5014bf4d9e56'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,

```

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

ANS: `db.shop.find({}, {product_name :1 ,product_material :1});`

```

> db.shop.find({}, {product_name :1 , product_material:1});
< {
  _id: ObjectId('666c325a1cca5014bf4d9e55'),
  product_name: 'Intelligent Fresh Chips',
  product_material: 'Concrete'
}
{
  _id: ObjectId('666c325a1cca5014bf4d9e56'),
  product_name: 'Practical Fresh Sausages',
  product_material: 'Cotton'
}
{
  _id: ObjectId('666c325a1cca5014bf4d9e57'),
  product_name: 'Refined Steel Car',

```

6. Find the product with a row id of 10

ANS: db.shop.find({id:"10"}):

```
db.shop.find({id : "10"});
{
  _id: ObjectId('666c325a1cca5014bf4d9e5e'),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 84,
  product_material: 'Frozen',
  product_color: 'indigo'
}
las atlas-sts14-shard-0 [primary] Task1> |
```

7. Find the product name and product material .

ANS: db.shop.find({}, {product_name :1 ,product_material :1});

```
> db.shop.find({}, {product_name :1 , product_material:1});
{
  {
    _id: ObjectId('666c325a1cca5014bf4d9e55'),
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  }
  {
    _id: ObjectId('666c325a1cca5014bf4d9e56'),
    product_name: 'Practical Fresh Sausages',
    product_material: 'Cotton'
  }
  {
    _id: ObjectId('666c325a1cca5014bf4d9e57'),
    product_name: 'Refined Steel Car',
    product_material: 'Soft'
  }
}
```

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

ANS: db.shop.find({product_material : 'Soft'});

```

db.shop.find({product_material: 'Soft'});
{
  _id: ObjectId('666c325a1cca5014bf4d9e58'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
}
{
  _id: ObjectId('666c325a1cca5014bf4d9e5d'),
  id: '9',
  product_name: 'Awesome Wooden Ball',
  product_price: 28,
  product_material: 'Soft',
  product_color: 'azure'
}
{
  _id: ObjectId('666c325a1cca5014bf4d9e5f'),
  id: '11',
  product_name: 'Unbranded Wooden Cheese',

```

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

ANS: `db.shop.find({$or :[{product_color:'indigo'},{product_price:492}] });`

```

db.shop.find({$or: [{product_color: 'indigo'}, {product_price: 492}] });

  _id: ObjectId('666c325a1cca5014bf4d9e56'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'

  _id: ObjectId('666c325a1cca5014bf4d9e58'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'

  _id: ObjectId('666c325a1cca5014bf4d9e5e'),
  id: '10',

```

10. Delete the products which product price value are 28

ANS: db.shop.deleteOne({product_price});

```

}
db.shop.deleteOne({product_price:28});
{
  acknowledged: true,
  deletedCount: 1
}

```