

---

Name : Prajakta Keer

Roll No : 33231

Class : TE 10

### SL1 ASSIGNMENT 13

Problem Statement : Execute at least 10 queries on above MongoDB database that demonstrates following querying techniques:

- Find
- FindOne (specific values)
- Conditional queries(Query conditionals, OR queries, \$not, Conditional semantics)
- Type-specific queries(Null, Regular expression, Querying arrays)

---

#### Find

```
> db.Orders.find().pretty();
{
  "_id" : ObjectId("5fc1d5bf7ba587b219205f0a"),
  "order_id" : 10,
  "prod_name" : "sanitizer",
  "quantity" : 15,
  "price" : 100,
  "delivery_date" : ISODate("2001-03-19T18:30:00Z")
}
{
  "_id" : ObjectId("5fc1d6b27ba587b219205f0e"),
  "order_id" : 17,
  "prod_name" : "wardrobe",
  "quantity" : 2,
  "price" : 7000,
  "delivery_date" : ISODate("2010-08-11T18:30:00Z")
}
```

#### Find One

```
> db.Orders.findOne({order_id : 10});
{
```

```

    "_id" : ObjectId("5fc1d5bf7ba587b219205f0a"),
    "order_id" : 10,
    "prod_name" : "sanitizer",
    "quantity" : 15,
    "price" : 100,
    "delivery_date" : ISODate("2001-03-19T18:30:00Z")
}

```

## Not

```

> db.Orders.find(quantity: {$not: {$gt: 5}}}).pretty();
2020-11-28T10:53:10.103+0530 E QUERY    [thread1] SyntaxError: missing ) after
argument list @(shell):1:23
> db.Orders.find({quantity: {$not: {$gt: 5}}}).pretty();
{
  "_id" : ObjectId("5fc1d6b27ba587b219205f0e"),
  "order_id" : 17,
  "prod_name" : "wardrobe",
  "quantity" : 2,
  "price" : 7000,
  "delivery_date" : ISODate("2010-08-11T18:30:00Z")
}

```

## Or

```

> db.Orders.find({ $or: [{prod_name : "wardrobe"}, {order_id : 10}]}).pretty();
{
  "_id" : ObjectId("5fc1d5bf7ba587b219205f0a"),
  "order_id" : 10,
  "prod_name" : "sanitizer",
  "quantity" : 15,
  "price" : 100,
  "delivery_date" : ISODate("2001-03-19T18:30:00Z")
}
{
  "_id" : ObjectId("5fc1d6b27ba587b219205f0e"),
  "order_id" : 17,
  "prod_name" : "wardrobe",
  "quantity" : 2,
  "price" : 7000,
  "delivery_date" : ISODate("2010-08-11T18:30:00Z")
}

```

## And

```

> db.Orders.find({ $and: [{prod_name : "wardrobe"}, {order_id : 17}]}).pretty();
{

```

```

    "_id" : ObjectId("5fc1d6b27ba587b219205f0e"),
    "order_id" : 17,
    "prod_name" : "wardrobe",
    "quantity" : 2,
    "price" : 7000,
    "delivery_date" : ISODate("2010-08-11T18:30:00Z")
}

```

## Greater than

```

> db.Orders.find({price : {$gt:99, $lt:7000}}).pretty();
{
  "_id" : ObjectId("5fc1d5bf7ba587b219205f0a"),
  "order_id" : 10,
  "prod_name" : "sanitizer",
  "quantity" : 15,
  "price" : 100,
  "delivery_date" : ISODate("2001-03-19T18:30:00Z")
}

```

## Greater than or equal to

```

> db.Orders.find({price : {$gte:100, $lt:7000}}).pretty();
{
  "_id" : ObjectId("5fc1d5bf7ba587b219205f0a"),
  "order_id" : 10,
  "prod_name" : "sanitizer",
  "quantity" : 15,
  "price" : 100,
  "delivery_date" : ISODate("2001-03-19T18:30:00Z")
}

```

## Returning Only Specific Fields

```

> db.Orders.find({price : 100},{prod_name : 1,quantity : 1}).pretty();
{
  "_id" : ObjectId("5fc1d5bf7ba587b219205f0a"),
  "prod_name" : "sanitizer",
  "quantity" : 15
}

```

## Regex(Find order whose prod\_name ends with 'er')

```

> db.Orders.find({prod_name:{ $regex:'er$'}}).pretty();
{
  "_id" : ObjectId("5fc1d5bf7ba587b219205f0a"),
  "order_id" : 10,
  "prod_name" : "sanitizer",
  "quantity" : 15,
  "price" : 100,
  "delivery_date" : ISODate("2001-03-19T18:30:00Z")
}

```