DAY – 35 QUERY'S

1.Find all the information about each products

```
db.find().pretty()
```

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

```
db.product.find({
   "product_price": {
      $gt: 399,
      $lt: 801
   }
}).pretty()
```

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

```
db.product.find({
   "product_price": {
      $not: {
         $lte: 600,
         $gte: 400
      }}}).pretty()
```

DAY – 35 QUERY'S

4.List the four products which are grater than 500 in price

```
db.product.find(
   {
      "product_price": {
         $gt: 50 }  }
).limit(4).pretty()
```

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

```
db.product.find({}, {
   product_material: 1,
   product_name: 1,
   _id: 0
})
```

6. Find the product with a row id of 10

```
db.product.find({
   id: '10'
})
```

DAY – 35 QUERY'S

7. Find only the product name and product material

```
db.product.find({}, {
   product_material: 1,
   product_name: 1,
   _id: 0})
```

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

```
db.product.find({
   product_material: {
      $in: ['Soft']   }})
```

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

```
db.product.find({
   product_price: 492,
   product_color: "Indigo"
})
```

DAY – 35 QUERY'S

10. Delete the products which product price value are same

```
db.product.aggregate([{$match: {}},
   {$group: {_id: { price: '$product_price'},
            product_ids: { $addToSet: "$id" },
            count: {$sum: 1}
           }
   },
   {$match:{
          count:{$gte: 2}
         }
   }
]).forEach(row => {
  print("Products Having duplicate prices: ", row.product_ids);
  row.product_ids.shift();
  print("Gonna remove the following Product ids: ", row.product_ids);
  db.product.remove({
     id: {
        $in: row.product_ids
     }  })});
```

DAY – 35 QUERY'S