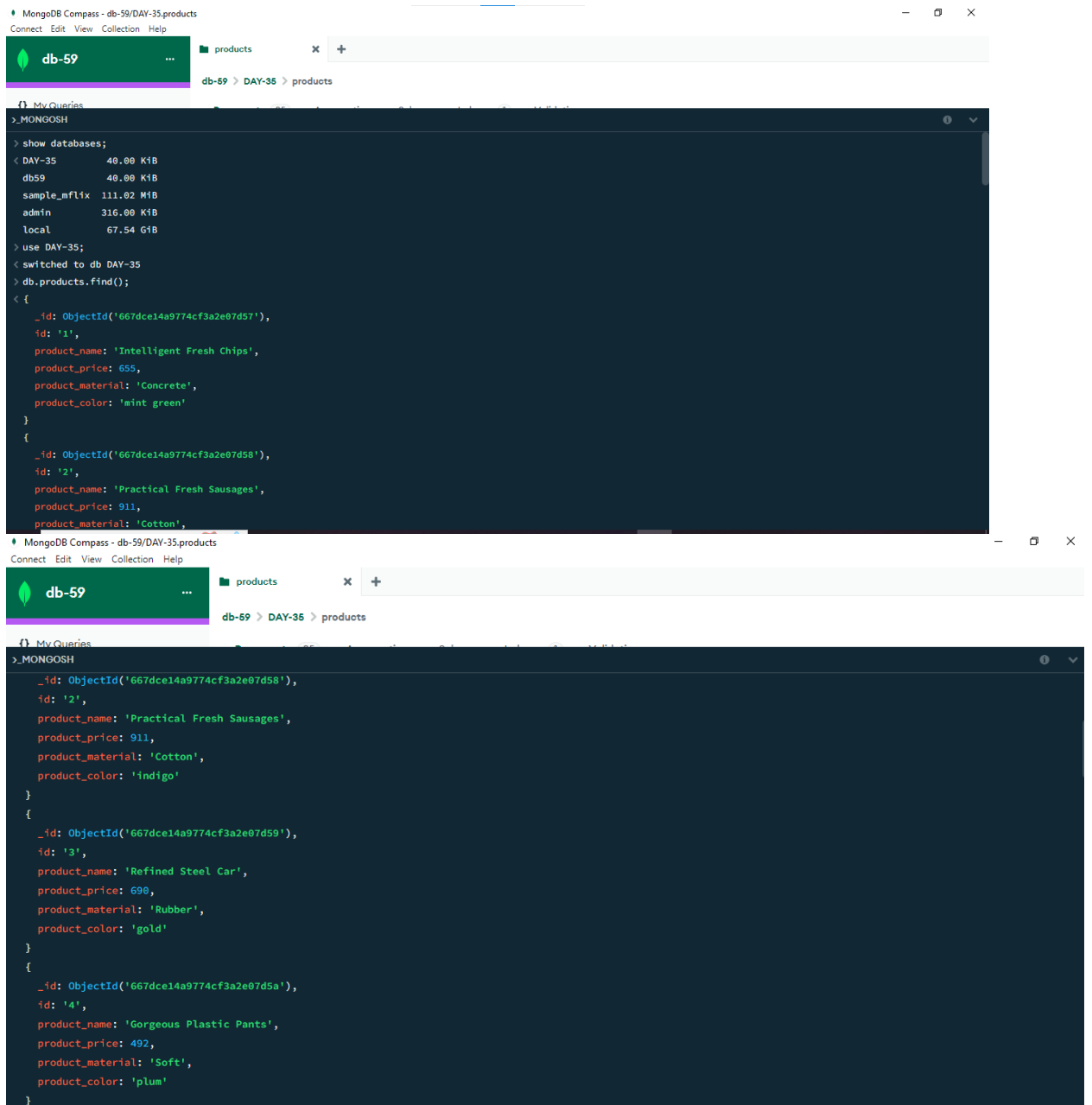


# GUVI TASK DAY-35 MONGODB TASK

For the following question write the corresponding MongoDB queries

1. Find all the information about each products

**db.product.find();**



The screenshot displays the MongoDB Compass interface. The top bar shows the connection to 'db-59' and the selected database 'DAY-35' with the 'products' collection. The 'My Queries' panel on the left shows the query `db.products.find()`. The main panel displays the results of the query, which are four product documents. Each document contains fields for `_id`, `id`, `product_name`, `product_price`, `product_material`, and `product_color`.

```
> show databases;
<
  DAY-35      40.00 KiB
  db59        40.00 KiB
  sample_mflix 111.02 MiB
  admin       316.00 KiB
  local       67.54 GiB
> use DAY-35;
< switched to db DAY-35
> db.products.find();
<
[
  {
    "_id": ObjectId("667dce14a9774cf3a2e07d57"),
    "id": "1",
    "product_name": "Intelligent Fresh Chips",
    "product_price": 655,
    "product_material": "Concrete",
    "product_color": "mint green"
  },
  {
    "_id": ObjectId("667dce14a9774cf3a2e07d58"),
    "id": "2",
    "product_name": "Practical Fresh Sausages",
    "product_price": 911,
    "product_material": "Cotton",
    "product_color": "indigo"
  },
  {
    "_id": ObjectId("667dce14a9774cf3a2e07d59"),
    "id": "3",
    "product_name": "Refined Steel Car",
    "product_price": 690,
    "product_material": "Rubber",
    "product_color": "gold"
  },
  {
    "_id": ObjectId("667dce14a9774cf3a2e07d5a"),
    "id": "4",
    "product_name": "Gorgeous Plastic Pants",
    "product_price": 492,
    "product_material": "Soft",
    "product_color": "plum"
  }
]
```

MongoDB Compass - db-59/DAY-35.products

Connect Edit View Collection Help

db-59

products

db-59 > DAY-35 > products

```
> MONGODB
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5b"),
  "id": "5",
  "product_name": "Sleek Cotton Chair",
  "product_price": 33,
  "product_material": "Fresh",
  "product_color": "black"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5c"),
  "id": "6",
  "product_name": "Awesome Wooden Towels",
  "product_price": 474,
  "product_material": "Plastic",
  "product_color": "orange"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5d"),
  "id": "7",
  "product_name": "Practical Soft Shoes",
  "product_price": 500,
  "product_material": "Rubber",
  "product_color": "pink"
}
```

MongoDB Compass - db-59/DAY-35.products

Connect Edit View Collection Help

db-59

products

db-59 > DAY-35 > products

```
> MONGODB
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5e"),
  "id": "8",
  "product_name": "Incredible Steel Hat",
  "product_price": 78,
  "product_material": "Rubber",
  "product_color": "violet"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5f"),
  "id": "9",
  "product_name": "Awesome Wooden Ball",
  "product_price": 28,
  "product_material": "Soft",
  "product_color": "azure"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d60"),
  "id": "10",
  "product_name": "Generic Wooden Pizza",
  "product_price": 84,
  "product_material": "Frozen",
  "product_color": "indigo"
}
```

MongoDB Compass - db-59/DAY-35.products

Connect Edit View Collection Help

db-59

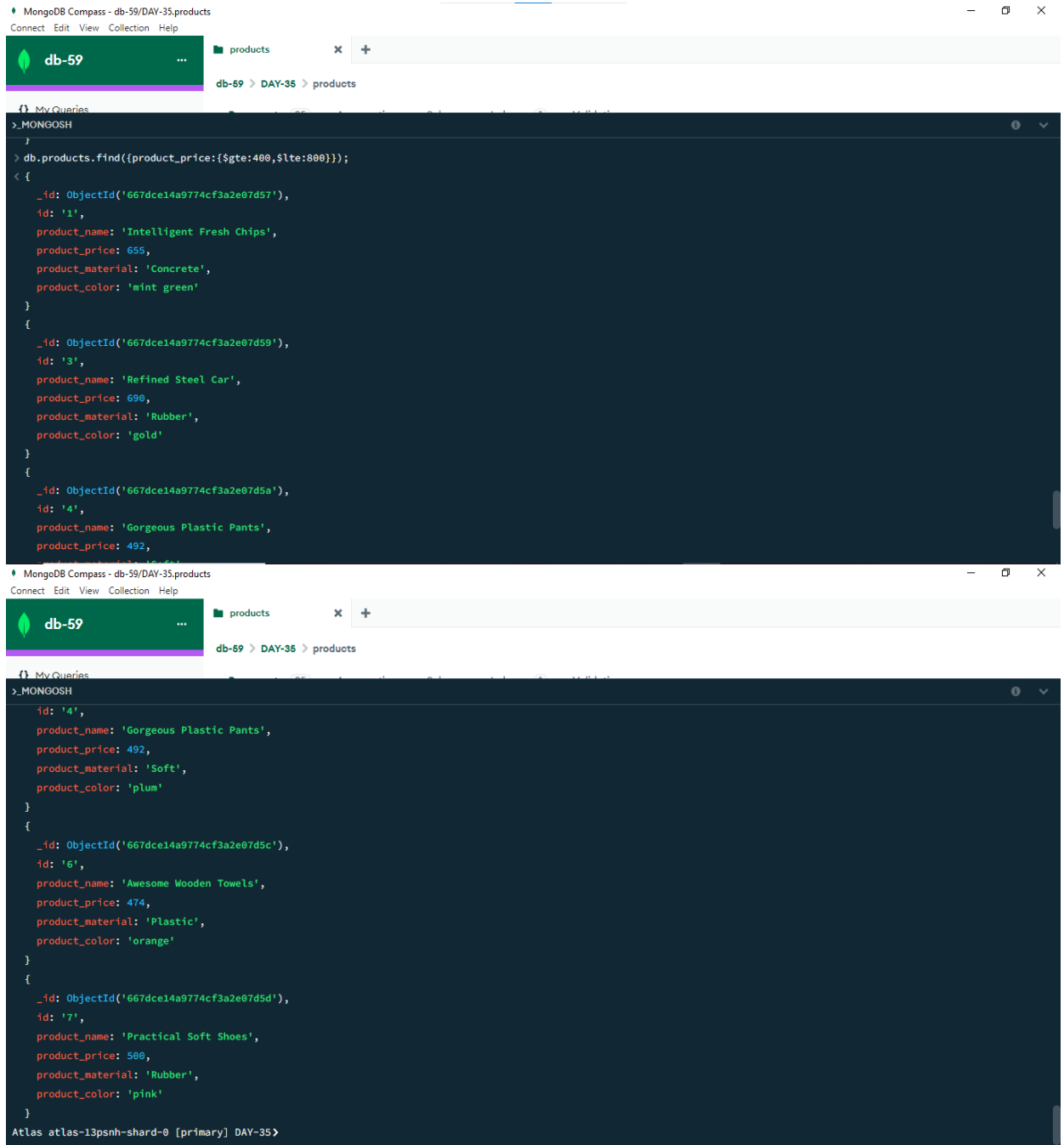
products

db-59 > DAY-35 > products

```
> MONGODB
{
  "_id": ObjectId("667dce14a9774cf3a2e07d6d"),
  "id": "23",
  "product_name": "Intelligent Steel Pizza",
  "product_price": 95,
  "product_material": "Cotton",
  "product_color": "azure"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d6e"),
  "id": "24",
  "product_name": "Tasty Rubber Cheese",
  "product_price": 47,
  "product_material": "Frozen",
  "product_color": "orchid"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d6f"),
  "id": "25",
  "product_name": "Licensed Steel Car",
  "product_price": 20,
  "product_material": "Cotton",
  "product_color": "indigo"
}
```

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

**db.product.find({product\_price: {\$gte: 400, \$lte: 800}});**



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection is to 'db-59' and the current view is 'products'. The query editor shows the following query:

```
>_MONGOOSH
> db.products.find({product_price: {$gte: 400, $lte: 800}});
```

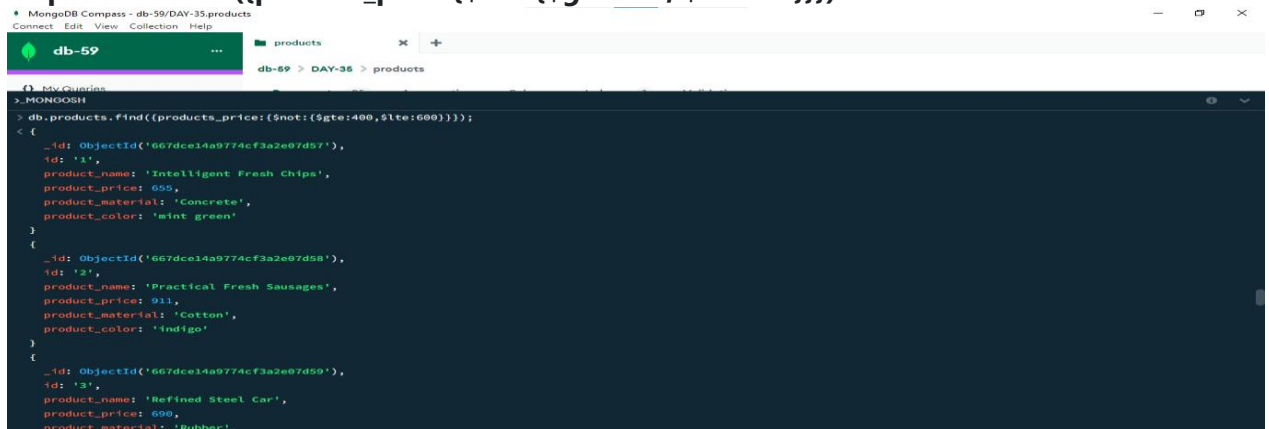
The results pane displays three documents:

```
{
  _id: ObjectId('667dce14a9774cf3a2e07d57'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d59'),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
  product_material: 'Rubber',
  product_color: 'gold'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d5a'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
}
```

The bottom of the screenshot shows the same interface with the results pane expanded, showing the full details of the documents, including the product name, price, material, and color. The status bar at the bottom indicates the connection is to 'Atlas atlas-13psnh-shard-0 [primary] DAY-35'.

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

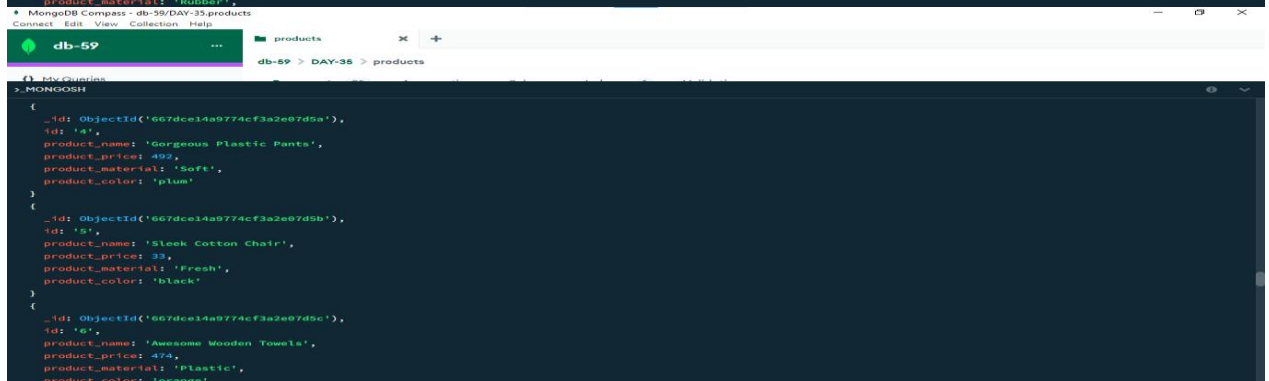
**db.products.find({product\_price:{\$not:{\$gte:400, \$lte:600}}})**



MongoDB Compass - db-59/DAY-35.products

db-59 > DAY-35 > products

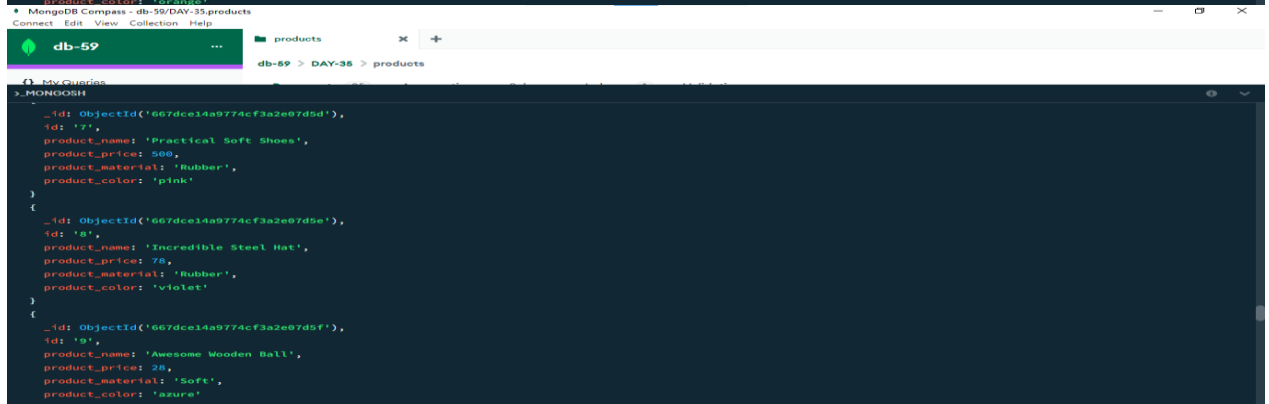
```
My Queries
> MONGOOSH
db.products.find({products_price:{$not:{$gte:400,$lte:600}}});
{
  "_id": ObjectId("667dce14a9774cf3a2e07d57"),
  "id": "1",
  "product_name": "Intelligent Fresh Chips",
  "product_price": 655,
  "product_material": "Concrete",
  "product_color": "mint green"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d58"),
  "id": "2",
  "product_name": "Practical Fresh Sausages",
  "product_price": 911,
  "product_material": "Cotton",
  "product_color": "indigo"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d59"),
  "id": "3",
  "product_name": "Refined Steel Car",
  "product_price": 690,
  "product_material": "Rubber",
  "product_color": "violet"
```



MongoDB Compass - db-59/DAY-35.products

db-59 > DAY-35 > products

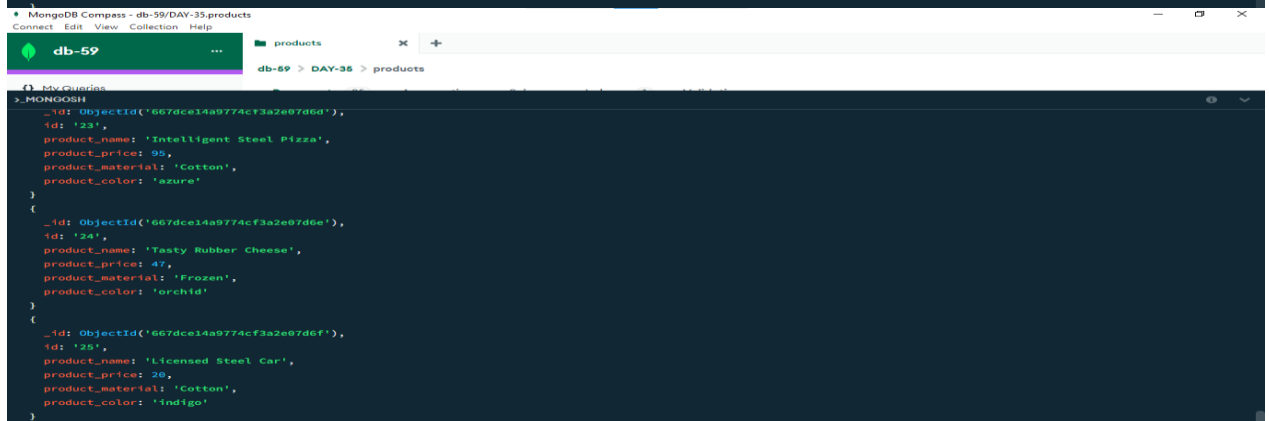
```
My Queries
> MONGOOSH
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5a"),
  "id": "4",
  "product_name": "Gorgeous Plastic Pants",
  "product_price": 492,
  "product_material": "Soft",
  "product_color": "plum"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5b"),
  "id": "5",
  "product_name": "Sleek Cotton Chair",
  "product_price": 35,
  "product_material": "Fresh",
  "product_color": "black"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5c"),
  "id": "6",
  "product_name": "Awesome Wooden Towels",
  "product_price": 474,
  "product_material": "Plastic",
  "product_color": "orange"
}
```



MongoDB Compass - db-59/DAY-35.products

db-59 > DAY-35 > products

```
My Queries
> MONGOOSH
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5d"),
  "id": "7",
  "product_name": "Practical Soft Shoes",
  "product_price": 500,
  "product_material": "Rubber",
  "product_color": "pink"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5e"),
  "id": "8",
  "product_name": "Incredible Steel Hat",
  "product_price": 76,
  "product_material": "Rubber",
  "product_color": "violet"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d5f"),
  "id": "9",
  "product_name": "Awesome Wooden Ball",
  "product_price": 28,
  "product_material": "Soft",
  "product_color": "azure"
}
```



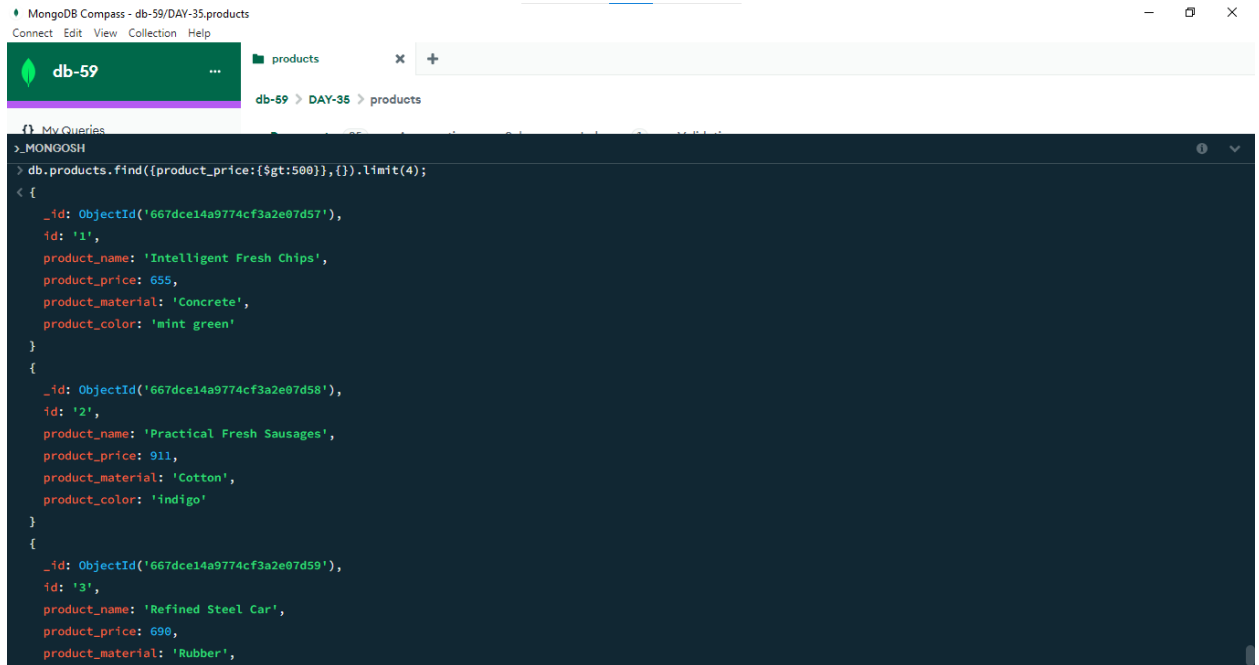
MongoDB Compass - db-59/DAY-35.products

db-59 > DAY-35 > products

```
My Queries
> MONGOOSH
{
  "_id": ObjectId("667dce14a9774cf3a2e07d6d"),
  "id": "23",
  "product_name": "Intelligent Steel Pizza",
  "product_price": 95,
  "product_material": "Cotton",
  "product_color": "azure"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d6e"),
  "id": "24",
  "product_name": "Tasty Rubber Cheese",
  "product_price": 47,
  "product_material": "Frozen",
  "product_color": "orchid"
}
{
  "_id": ObjectId("667dce14a9774cf3a2e07d6f"),
  "id": "25",
  "product_name": "Licensed Steel Car",
  "product_price": 20,
  "product_material": "Cotton",
  "product_color": "indigo"
}
```

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

**db.products.find({product\_price:{\$gt:500}},{}).limit(4)**

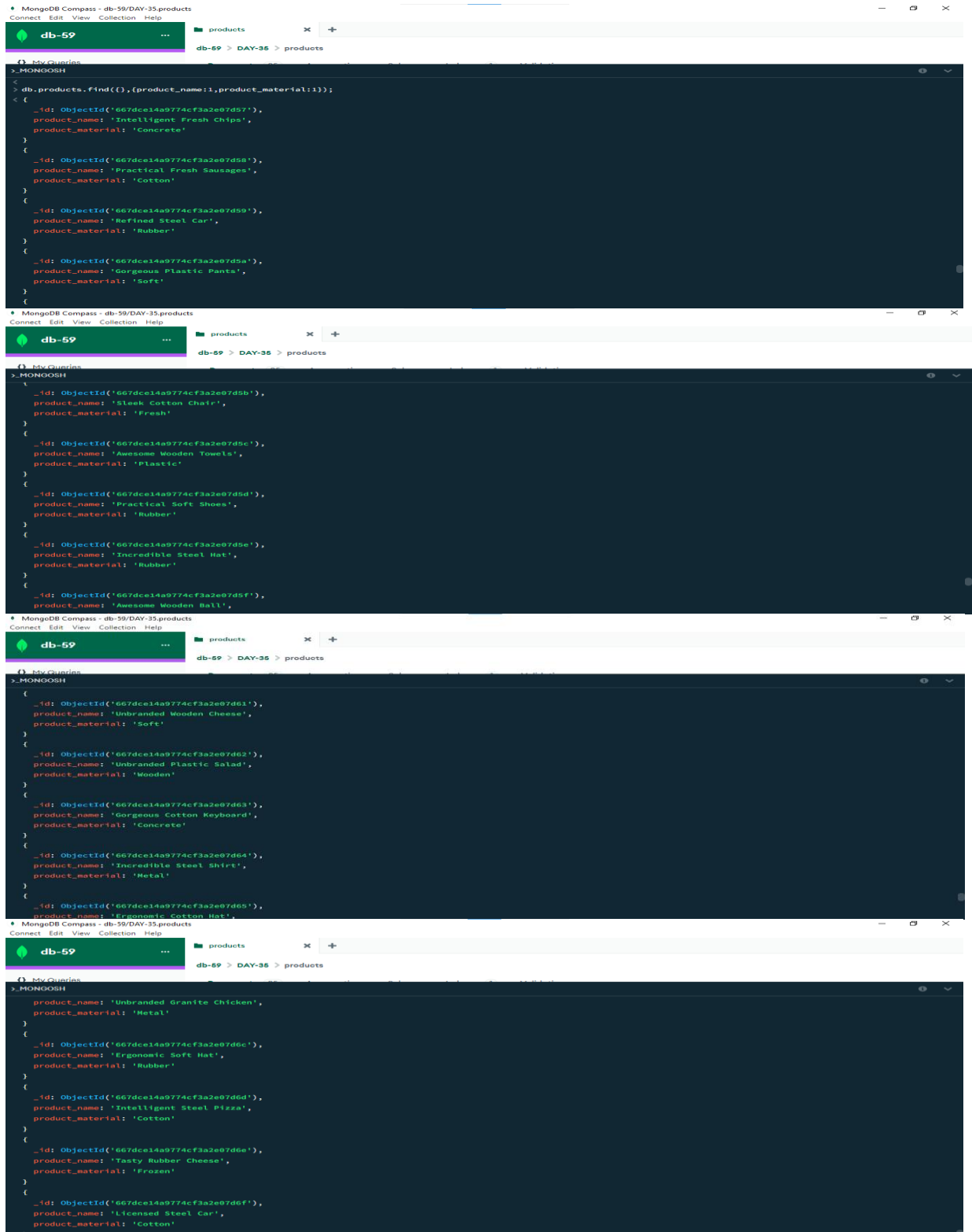


The screenshot shows the MongoDB Compass application. The top bar indicates the connection to 'db-59/DAV-35.products'. The left sidebar shows the 'My Queries' tab. The main area displays the query `db.products.find({product_price:{$gt:500}},{}).limit(4);` and its results. The results are a JSON array of three objects (the fourth is truncated).

```
>_MONGOOSH
> db.products.find({product_price:{$gt:500}},{}).limit(4);
< [
  {
    _id: ObjectId('667dce14a9774cf3a2e07d57'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('667dce14a9774cf3a2e07d58'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('667dce14a9774cf3a2e07d59'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'indigo'
  },
  ...
]
```

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

```
db.products.find({}, {product_name:1, product_material:1})
```

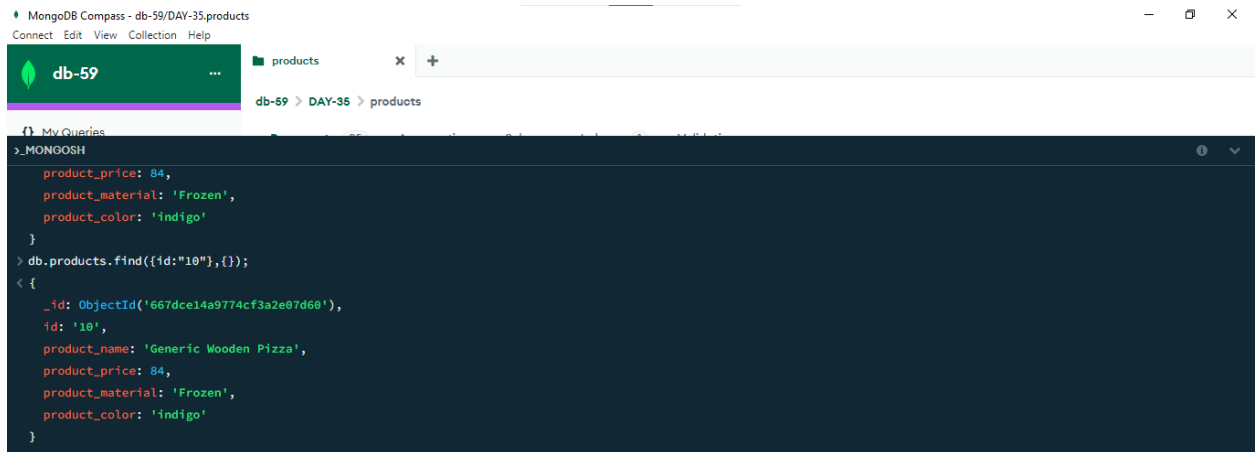


The following table represents the data shown in the screenshots:

product_name	product_material
Intelligent Fresh Chips	Concrete
Practical Fresh Sausages	Cotton
Gorgeous Plastic Pants	Rubber
Sleek Cotton Chair	Fresh
Awesome Wooden Towels	Plastic
Practical Soft Shoes	Rubber
Incredible Steel Hat	Rubber
Awesome Wooden Ball	
Unbranded Wooden Cheese	Soft
Unbranded Plastic Salad	Wooden
Gorgeous Cotton Keyboard	Concrete
Incredible Steel Shirt	Metal
Unbranded Granite Chicken	Metal
Ergonomic Soft Hat	Rubber
Intelligent Steel Pizza	Cotton
Tasty Rubber Cheese	Frozen
Licensed Steel Car	Cotton

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

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

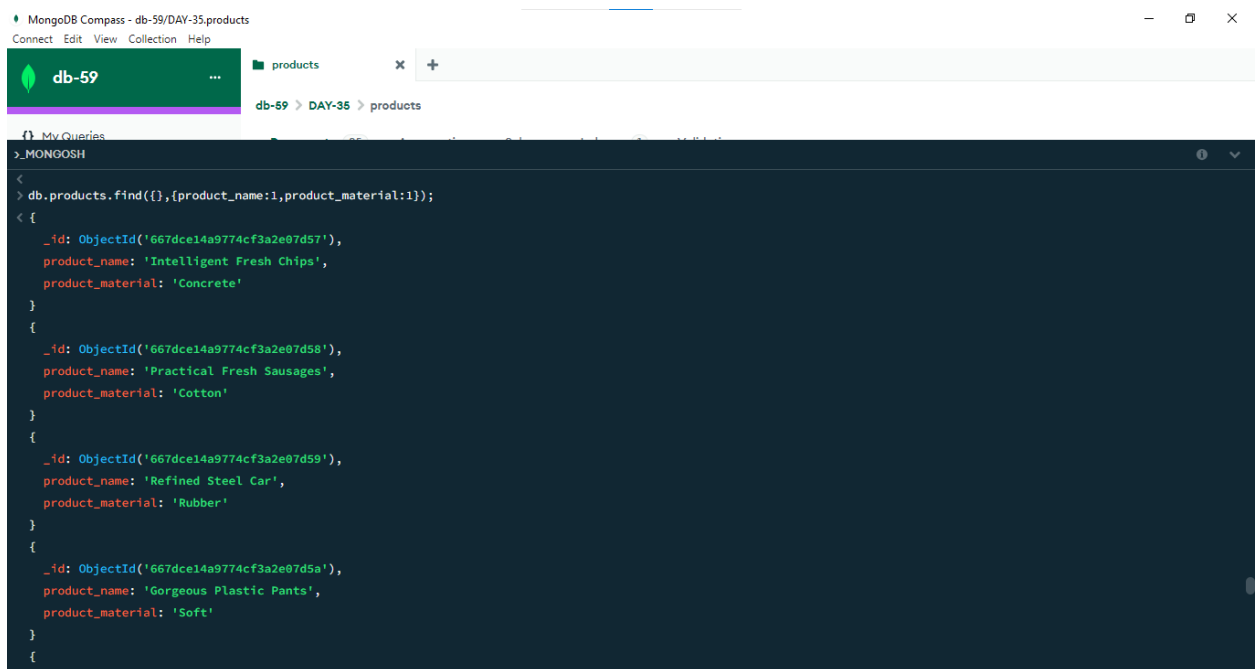


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection is 'db-59/DAV-35.products'. The left sidebar shows the 'db-59' database and the 'products' collection. The main editor area contains the following query:

```
>_MONGOOSH
product_price: 84,
product_material: 'Frozen',
product_color: 'Indigo'
}
db.products.find({id:"10"},{});
< {
  _id: ObjectId('667dce14a9774cf3a2e07d60'),
  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.

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

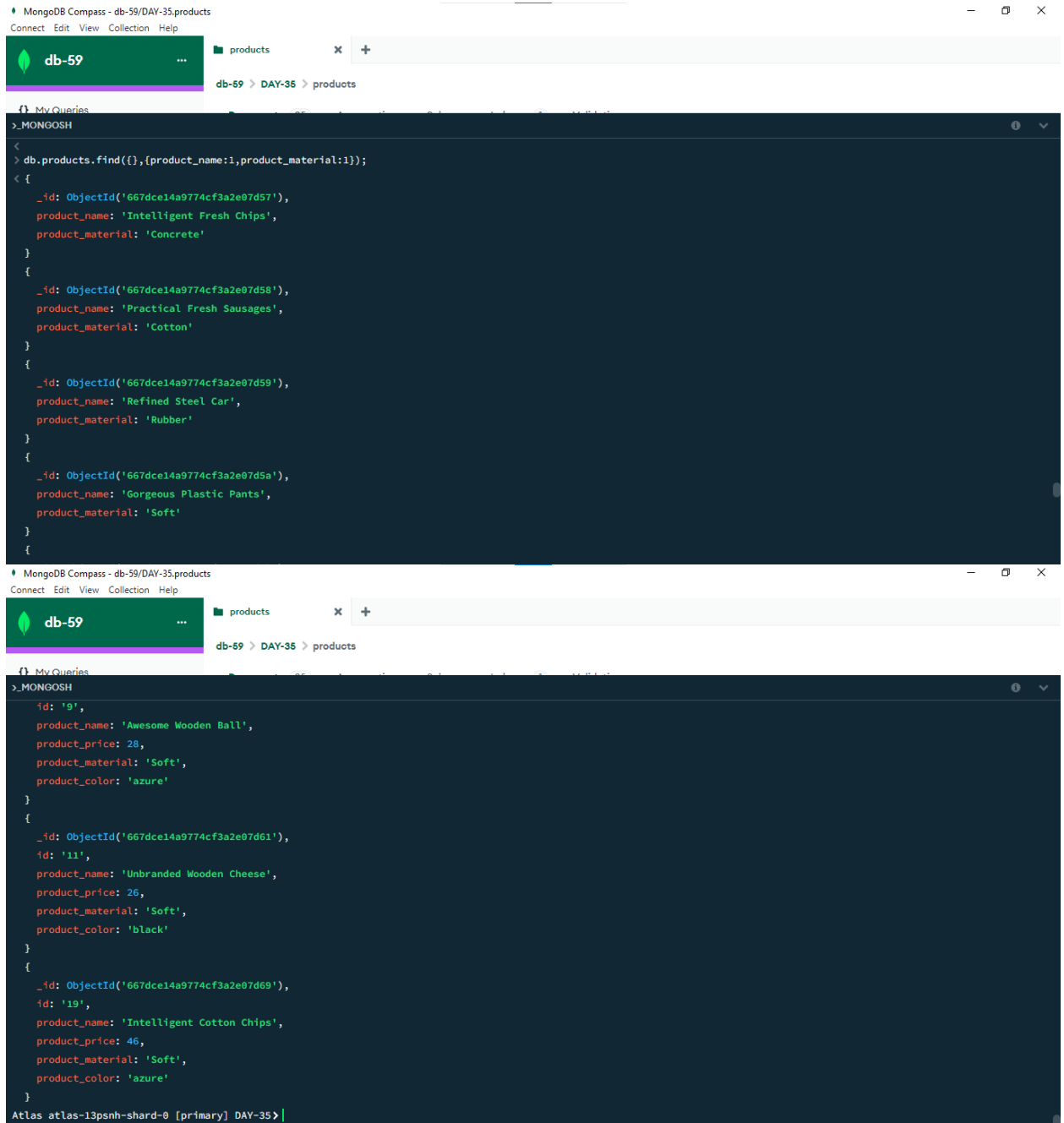


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection is 'db-59/DAV-35.products'. The left sidebar shows the 'db-59' database and the 'products' collection. The main editor area contains the following query:

```
>_MONGOOSH
db.products.find({}, {product_name:1, product_material:1});
< {
  _id: ObjectId('667dce14a9774cf3a2e07d57'),
  product_name: 'Intelligent Fresh Chips',
  product_material: 'Concrete'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d58'),
  product_name: 'Practical Fresh Sausages',
  product_material: 'Cotton'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d59'),
  product_name: 'Refined Steel Car',
  product_material: 'Rubber'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d5a'),
  product_name: 'Gorgeous Plastic Pants',
  product_material: 'Soft'
}
{
```

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

**db.products.find({product\_material:"Soft"})**



The screenshot shows the MongoDB Compass interface. The top bar indicates the database is 'db-59' and the collection is 'products'. The query editor shows the following query:

```
> db.products.find({}, {product_name:1, product_material:1});
```

The results are displayed in a JSON array format:

```
< {
  _id: ObjectId('667dce14a9774cf3a2e07d57'),
  product_name: 'Intelligent Fresh Chips',
  product_material: 'Concrete'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d58'),
  product_name: 'Practical Fresh Sausages',
  product_material: 'Cotton'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d59'),
  product_name: 'Refined Steel Car',
  product_material: 'Rubber'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d5a'),
  product_name: 'Gorgeous Plastic Pants',
  product_material: 'Soft'
}
}
```

The bottom part of the screenshot shows the same interface with a different query:

```
> db.products.find({product_material:"Soft"});
```

The results are displayed in a JSON array format:

```
< {
  id: '9',
  product_name: 'Awesome Wooden Ball',
  product_price: 28,
  product_material: 'Soft',
  product_color: 'azure'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d61'),
  id: '11',
  product_name: 'Unbranded Wooden Cheese',
  product_price: 26,
  product_material: 'Soft',
  product_color: 'black'
}
{
  _id: ObjectId('667dce14a9774cf3a2e07d69'),
  id: '19',
  product_name: 'Intelligent Cotton Chips',
  product_price: 46,
  product_material: 'Soft',
  product_color: 'azure'
}
}
```

The status bar at the bottom indicates the connection is to 'Atlas atlas-13psnh-shard-0 [primary] DAY-35'.



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

**db.products.find({product\_color:"indigo",product\_price:492.00})**

```
    product_price: 28
  }
> db.products.find({product_color:"indigo",product_price:492.00});
<
Atlas atlas-13psnh-shard-0 [primary] DAY-35>
```

10. Delete the products which product price value are 28.

**db.products.deleteOne({product\_price:28})**

```
<
> db.products.deleteOne({product_price:28});
< {
  acknowledged: true,
  deletedCount: 1
}
Atlas atlas-13psnh-shard-0 [primary] DAY-35>
```