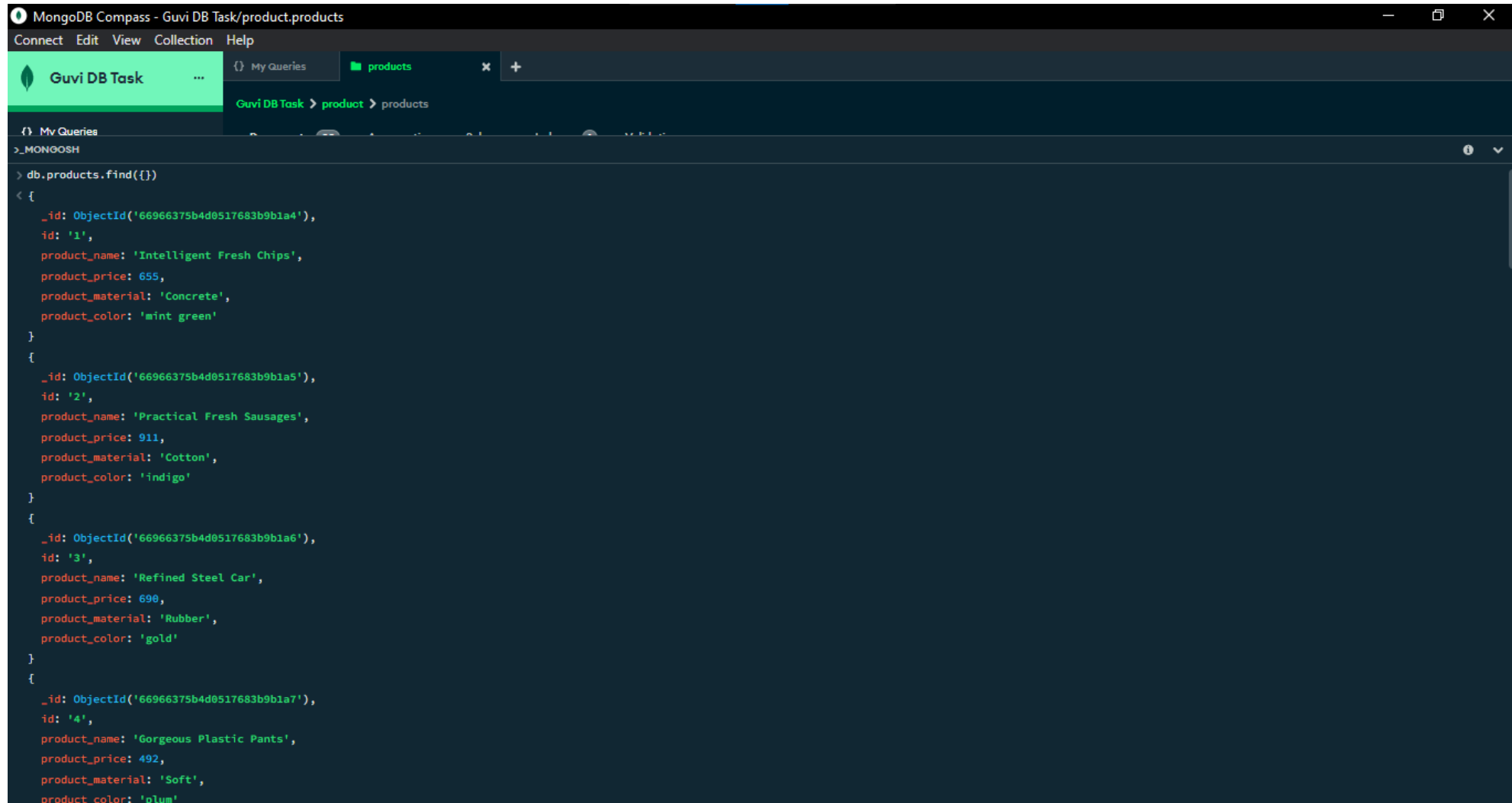


MongoDB Task

1. Find all the information about each products

Query: `db.products.find({})`

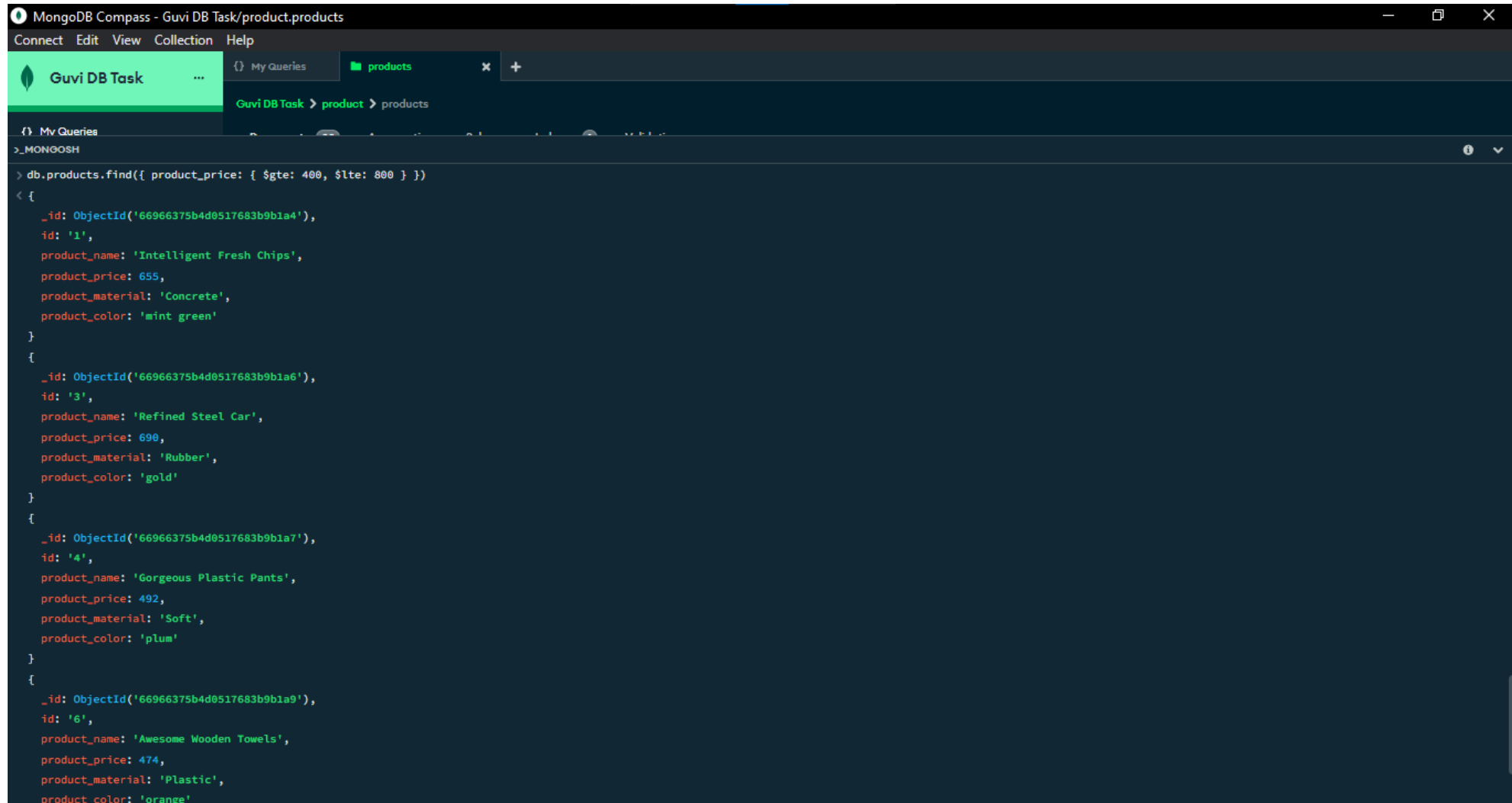


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'Guvu DB Task/product.products'. The left sidebar shows the 'Guvu DB Task' database and the 'products' collection. The main area displays the query `db.products.find({})` and its results. The results are a JSON array of four product documents.

```
> db.products.find({})
< [
  {
    _id: ObjectId('66966375b4d0517683b9b1a4'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a5'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a6'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a7'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  }
]
```

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

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

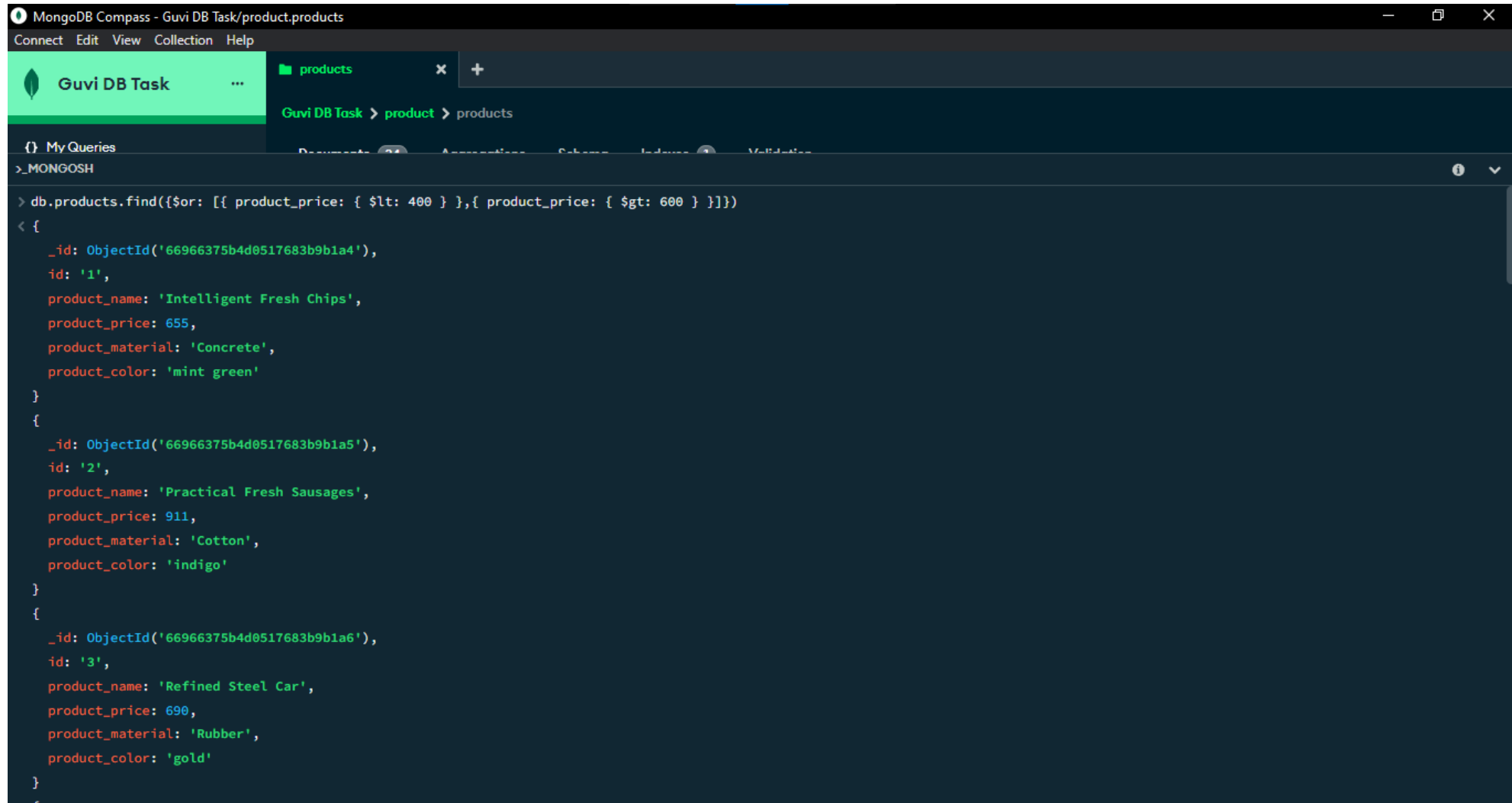


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection is to 'Guvu DB Task/product.products'. The left sidebar shows the 'Guvu DB Task' database selected. The main panel displays a query: `db.products.find({ product_price: { $gte: 400, $lte: 800 } })`. The results are shown as a JSON array with four objects, each representing a product. The products are: 'Intelligent Fresh Chips' (price 655), 'Refined Steel Car' (price 690), 'Gorgeous Plastic Pants' (price 492), and 'Awesome Wooden Towels' (price 474).

```
> db.products.find({ product_price: { $gte: 400, $lte: 800 } })
< [
  {
    _id: ObjectId('66966375b4d0517683b9b1a4'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a6'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a7'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a9'),
    id: '6',
    product_name: 'Awesome Wooden Towels',
    product_price: 474,
    product_material: 'Plastic',
    product_color: 'orange'
  }
]
```

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

Query: `db.products.find({ $or: [{ product_price: { $lt: 400 } }, { product_price: { $gt: 600 } }] })`



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'Guvi DB Task/product.products'. The left sidebar shows the 'Guvi DB Task' database and the 'products' collection. The main area displays a query in the 'My Queries' tab:

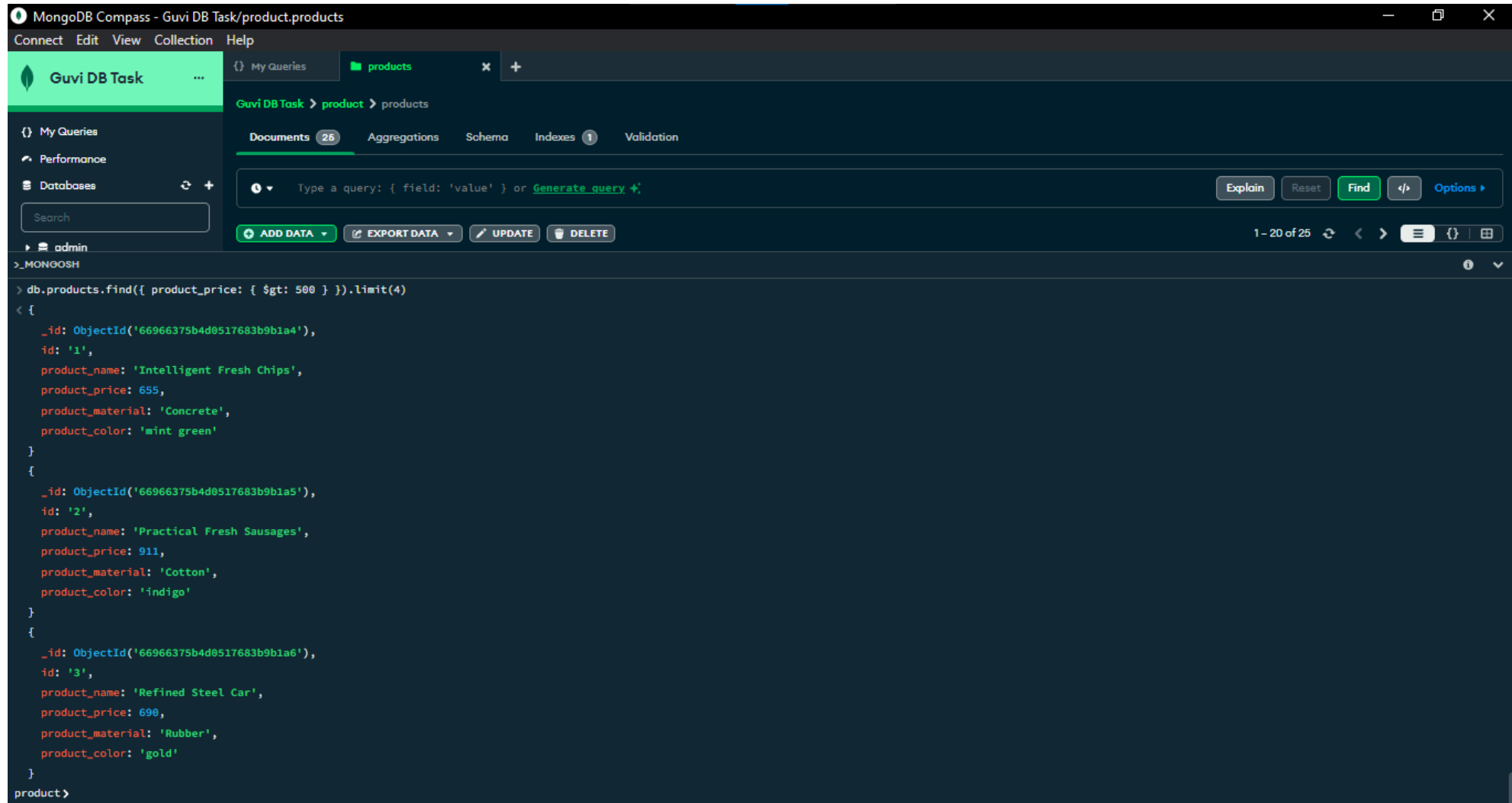
```
> db.products.find({$or: [{ product_price: { $lt: 400 } },{ product_price: { $gt: 600 } }]}))
```

The results are shown in a table with 3 documents:

id	product_name	product_price	product_material	product_color
1	Intelligent Fresh Chips	655	Concrete	mint green
2	Practical Fresh Sausages	911	Cotton	indigo
3	Refined Steel Car	690	Rubber	gold

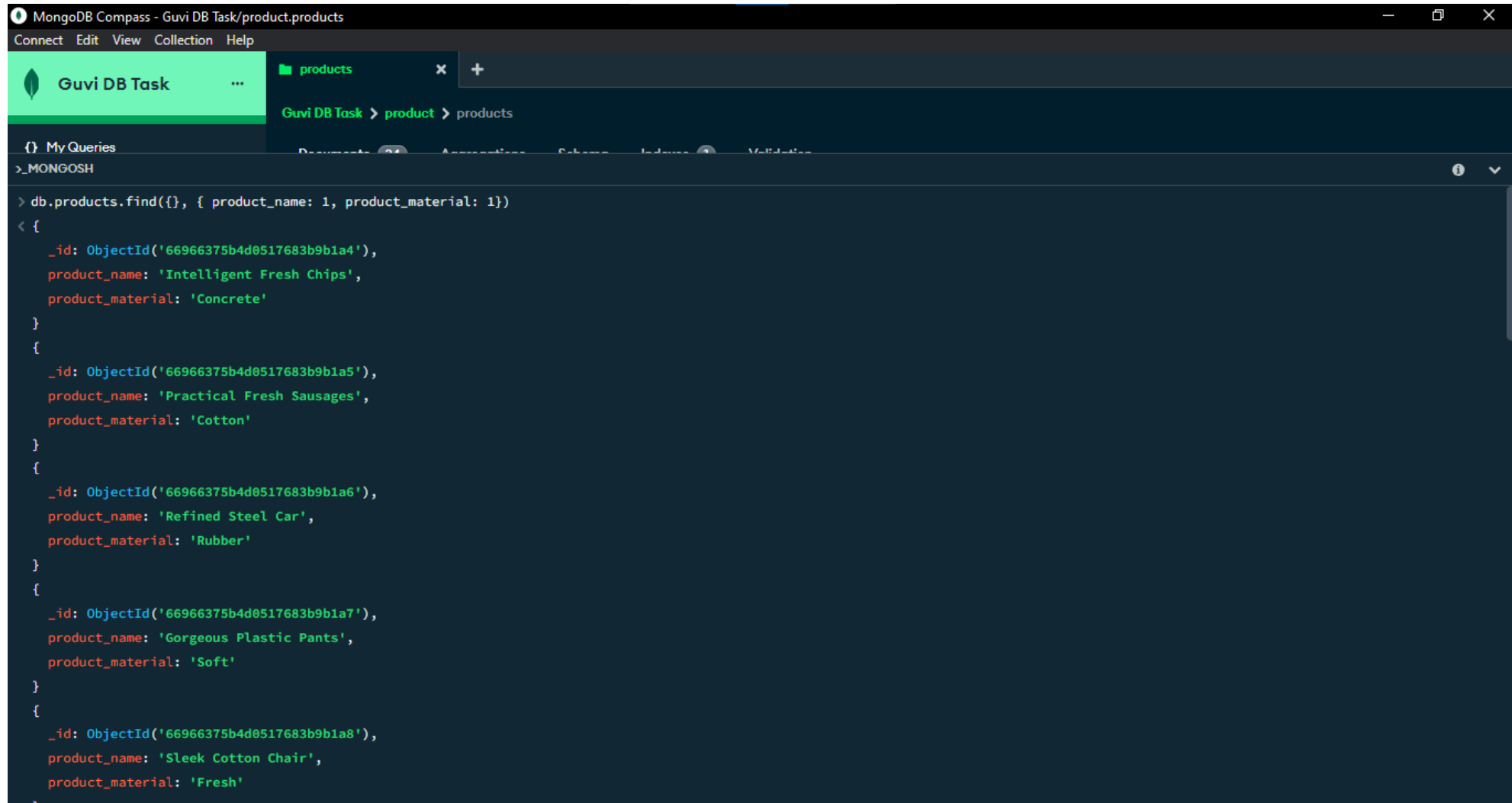
4. List the four product which are greater than 500 in price

Query: `db.products.find({ product_price: { $gt: 500 } }).limit(4)`



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

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

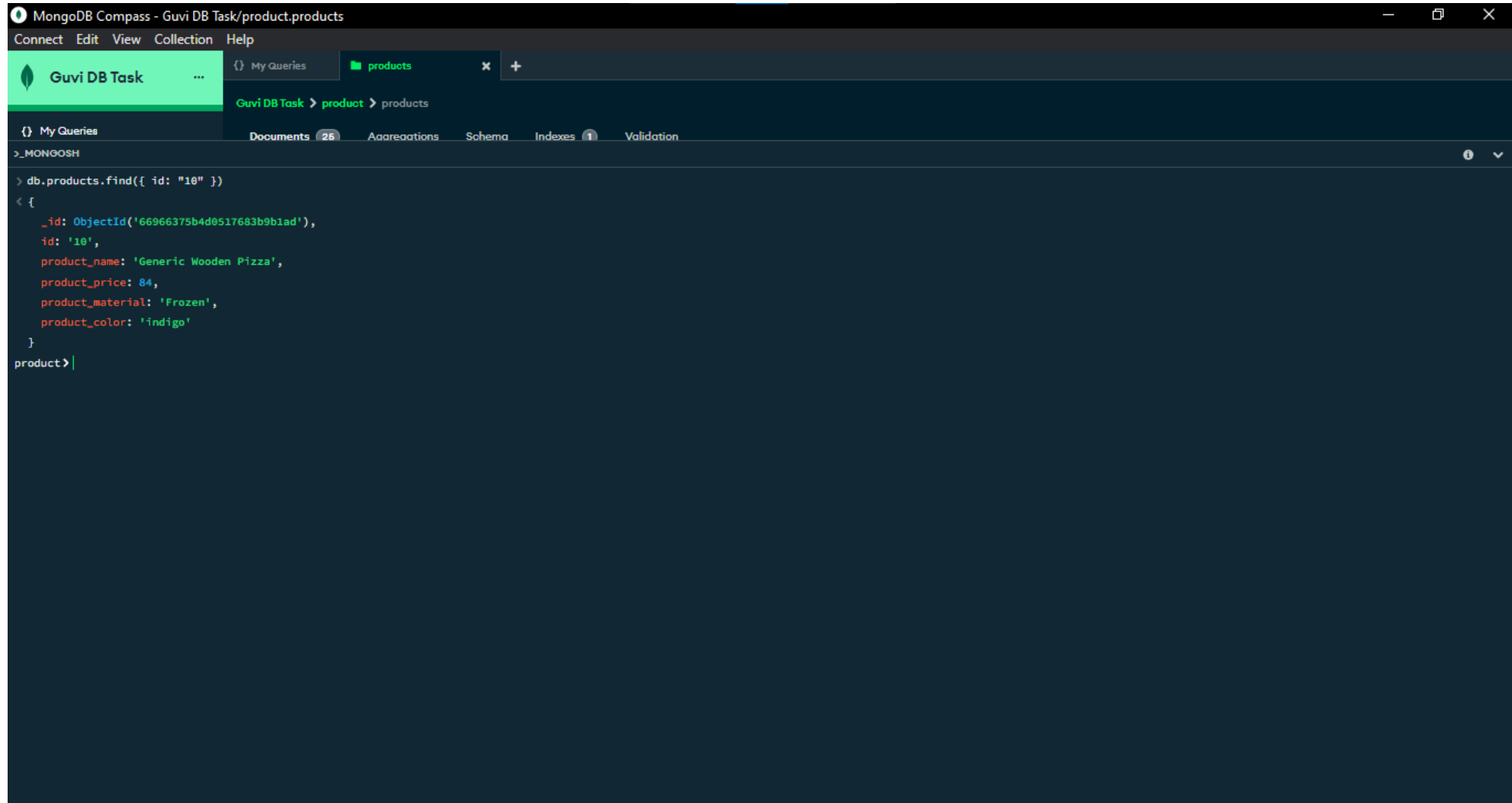


The screenshot shows the MongoDB Compass interface. The title bar reads "MongoDB Compass - Guvi DB Task/product.products". The left sidebar has a "Guvi DB Task" button. The main area shows a breadcrumb "Guvi DB Task > product > products". Below this is a "My Queries" tab with a "MONGOSH" prompt. The query entered is `db.products.find({}, { product_name: 1, product_material: 1})`. The results are displayed as a JSON array of five objects, each containing `_id`, `product_name`, and `product_material`.

```
> db.products.find({}, { product_name: 1, product_material: 1})
< [
  {
    _id: ObjectId('66966375b4d0517683b9b1a4'),
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a5'),
    product_name: 'Practical Fresh Sausages',
    product_material: 'Cotton'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a6'),
    product_name: 'Refined Steel Car',
    product_material: 'Rubber'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a7'),
    product_name: 'Gorgeous Plastic Pants',
    product_material: 'Soft'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a8'),
    product_name: 'Sleek Cotton Chair',
    product_material: 'Fresh'
  }
]
```

6. Find the product with a row id of 10

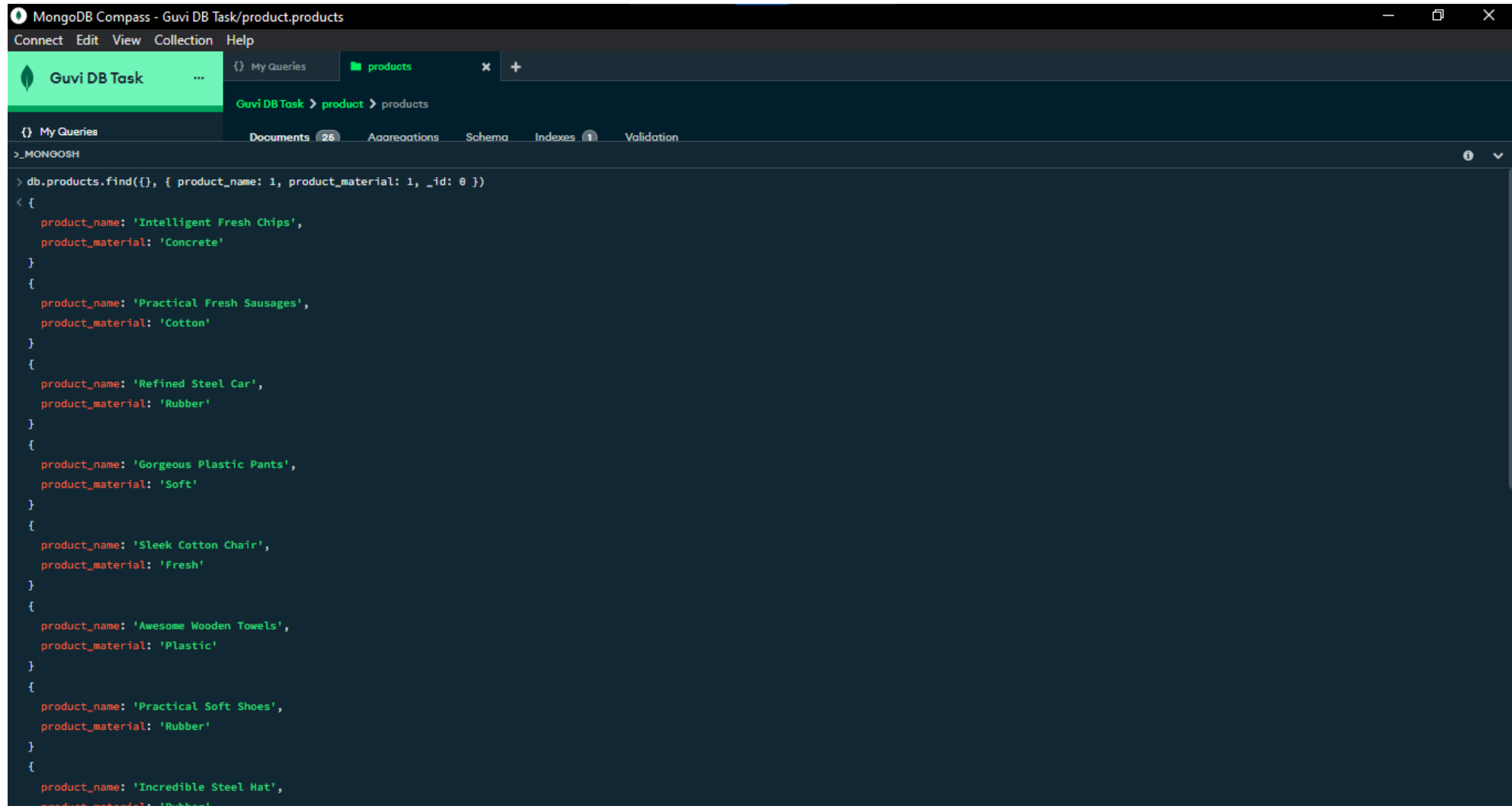
Query: `db.products.find({ id: "10" })`



The screenshot shows the MongoDB Compass interface. The title bar reads "MongoDB Compass - Guvi DB Task/product.products". The menu bar includes "Connect", "Edit", "View", "Collection", and "Help". A sidebar on the left shows "Guvi DB Task" with a dropdown arrow. The main panel has a tab for "products" and a breadcrumb "Guvi DB Task > product > products". Below the breadcrumb are tabs for "Documents" (26), "Aggregations", "Schema", "Indexes" (1), and "Validation". The "Documents" tab is active, showing a MongoDB shell prompt ">_MONGOSSH". The query entered is `> db.products.find({ id: "10" })`. The result is a single document: `< { _id: ObjectId('66966375b4d9517683b9blad'), id: '10', product_name: 'Generic Wooden Pizza', product_price: 84, product_material: 'Frozen', product_color: 'indigo' }`. The prompt `product>` is visible at the bottom.

7. Find only the product name and product material

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

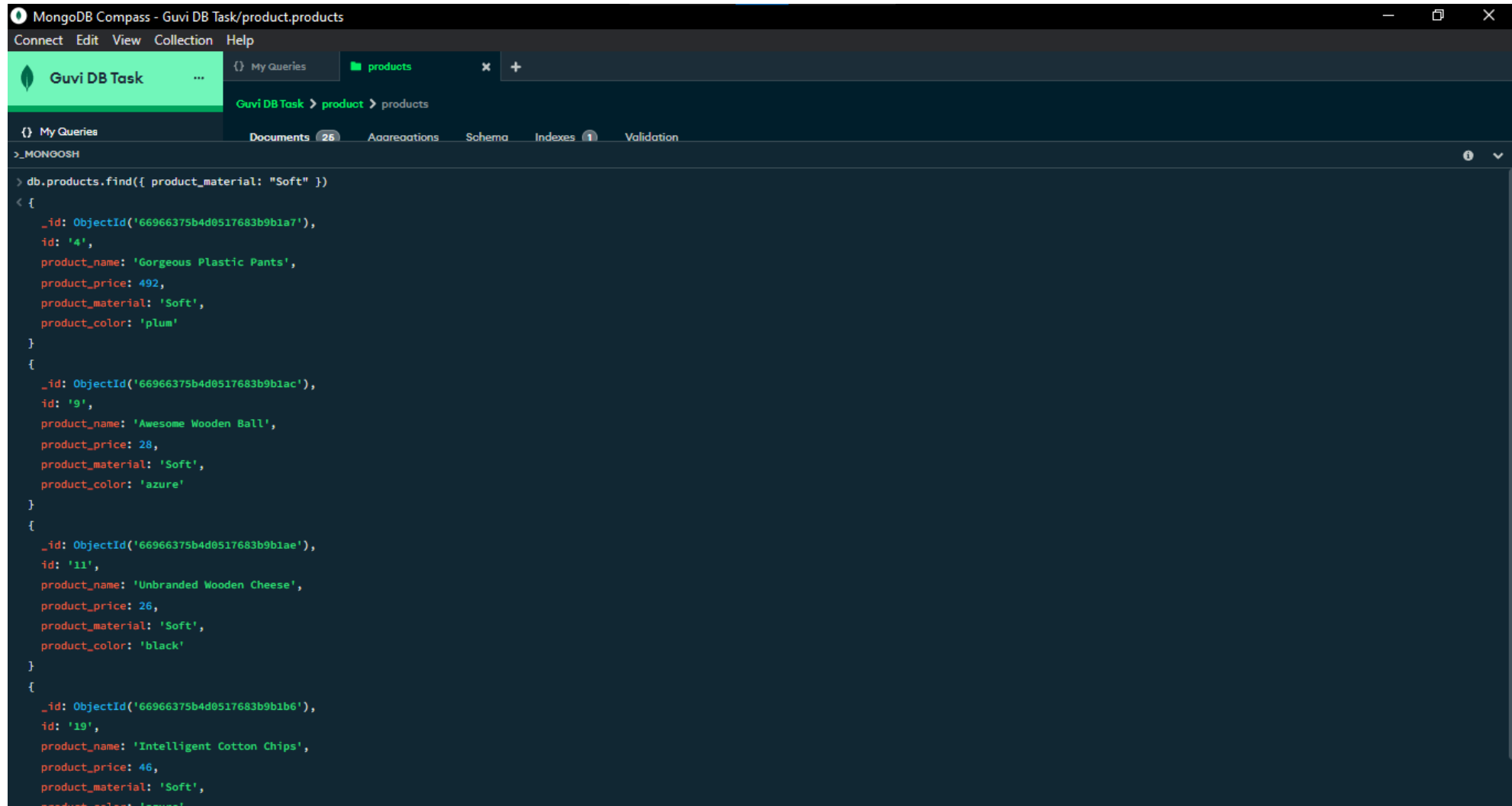


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'Guvi DB Task/product.products'. The left sidebar shows the 'Guvi DB Task' database and the 'products' collection. The main panel displays the query `db.products.find({}, { product_name: 1, product_material: 1, _id: 0 })` and its results. The results are a list of 10 documents, each containing 'product_name' and 'product_material'.

```
> db.products.find({}, { product_name: 1, product_material: 1, _id: 0 })
< {
  product_name: 'Intelligent Fresh Chips',
  product_material: 'Concrete'
}
{
  product_name: 'Practical Fresh Sausages',
  product_material: 'Cotton'
}
{
  product_name: 'Refined Steel Car',
  product_material: 'Rubber'
}
{
  product_name: 'Gorgeous Plastic Pants',
  product_material: 'Soft'
}
{
  product_name: 'Sleek Cotton Chair',
  product_material: 'Fresh'
}
{
  product_name: 'Awesome Wooden Towels',
  product_material: 'Plastic'
}
{
  product_name: 'Practical Soft Shoes',
  product_material: 'Rubber'
}
{
  product_name: 'Incredible Steel Hat',
  product_material: 'Rubber'
}
```


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

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

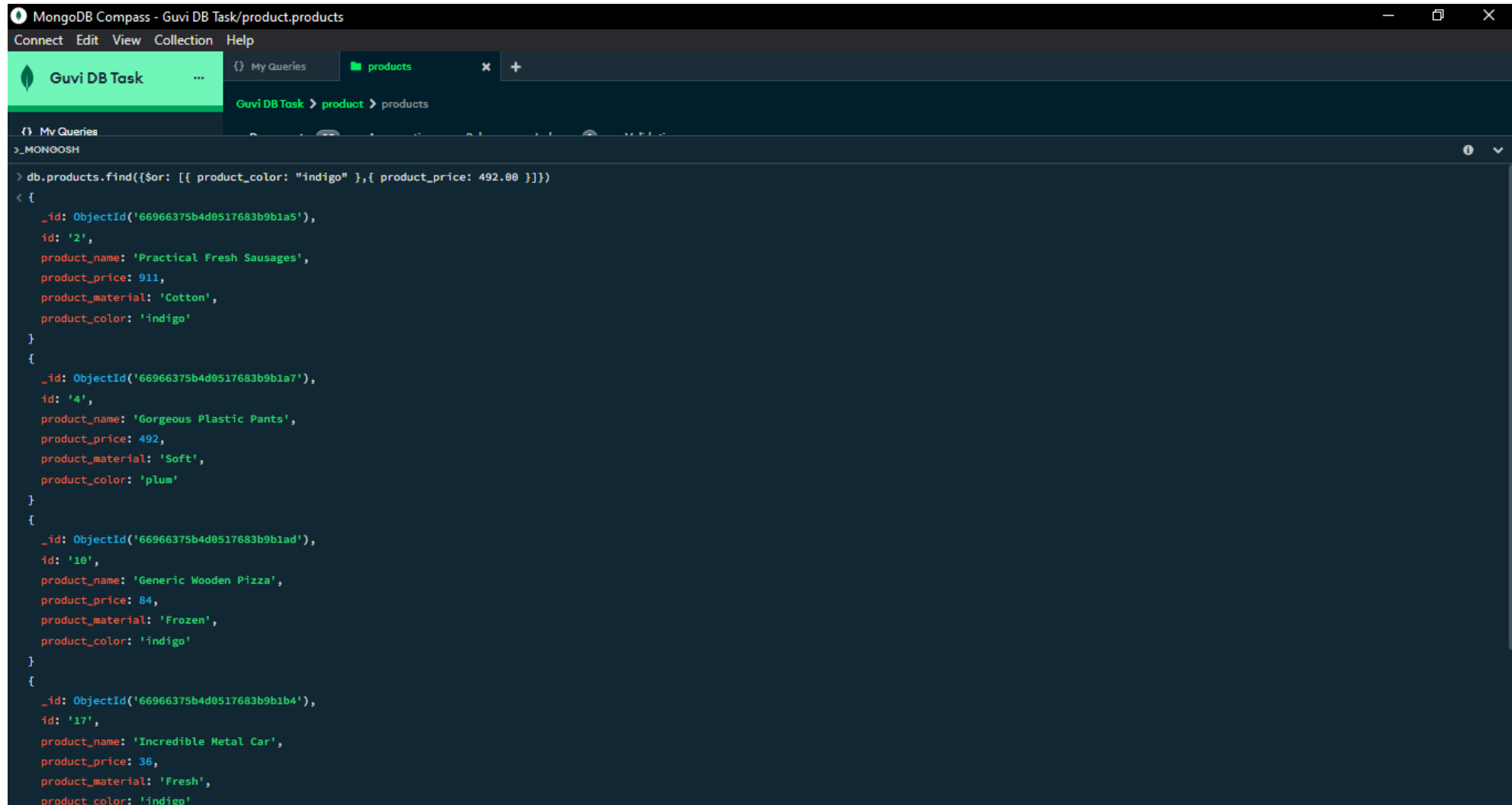


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection is to 'Guvu DB Task/product.products'. The left sidebar shows the 'Guvu DB Task' database selected. The main panel displays the 'products' collection with 26 documents. The query `db.products.find({ product_material: "Soft" })` is entered in the command line, and the results are shown as a JSON array of four product documents.

```
> db.products.find({ product_material: "Soft" })
< [
  {
    _id: ObjectId('66966375b4d0517683b9b1a7'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1ac'),
    id: '9',
    product_name: 'Awesome Wooden Ball',
    product_price: 28,
    product_material: 'Soft',
    product_color: 'azure'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1ae'),
    id: '11',
    product_name: 'Unbranded Wooden Cheese',
    product_price: 26,
    product_material: 'Soft',
    product_color: 'black'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1b6'),
    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

Query: `db.products.find({ $or: [{ product_color: "indigo" }, { product_price: 492.00 }] })`



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'Guvi DB Task/product.products'. The left sidebar shows the 'Guvi DB Task' database and the 'products' collection. The main area displays the following query:

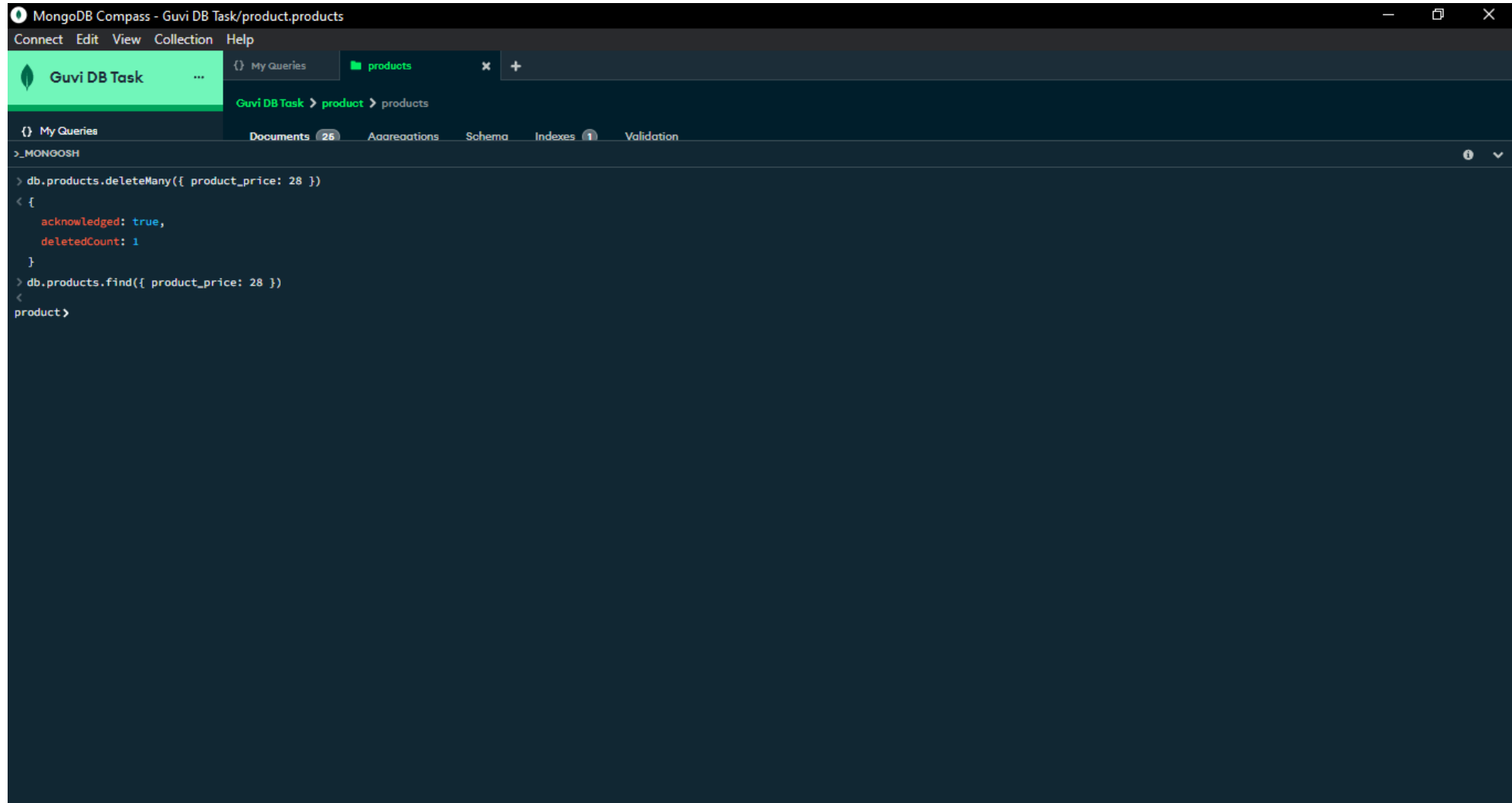
```
> db.products.find({$or: [{ product_color: "indigo" }, { product_price: 492.00 }]}])
```

The results are displayed as a JSON array with four objects:

```
< [
  {
    _id: ObjectId('66966375b4d0517683b9b1a5'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1a7'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1ad'),
    id: '10',
    product_name: 'Generic Wooden Pizza',
    product_price: 84,
    product_material: 'Frozen',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('66966375b4d0517683b9b1b4'),
    id: '17',
    product_name: 'Incredible Metal Car',
    product_price: 36,
    product_material: 'Fresh',
    product_color: 'indigo'
  }
]
```

10. Delete the products which product price value are 28

Query: `db.products.deleteMany({ product_price: 28 })`



The screenshot shows the MongoDB Compass application window titled "MongoDB Compass - Guvi DB Task/product.products". The interface includes a top menu bar with "Connect", "Edit", "View", "Collection", and "Help". Below the menu is a sidebar with a green "Guvi DB Task" button and a "My Queries" tab. The main area displays the "products" collection, with a breadcrumb path "Guvi DB Task > product > products". The "Documents" tab is active, showing 25 documents. The command prompt area contains the following text:

```
> _MONGOSSH
> db.products.deleteMany({ product_price: 28 })
< {
  acknowledged: true,
  deletedCount: 1
}
> db.products.find({ product_price: 28 })
<
product>
```