

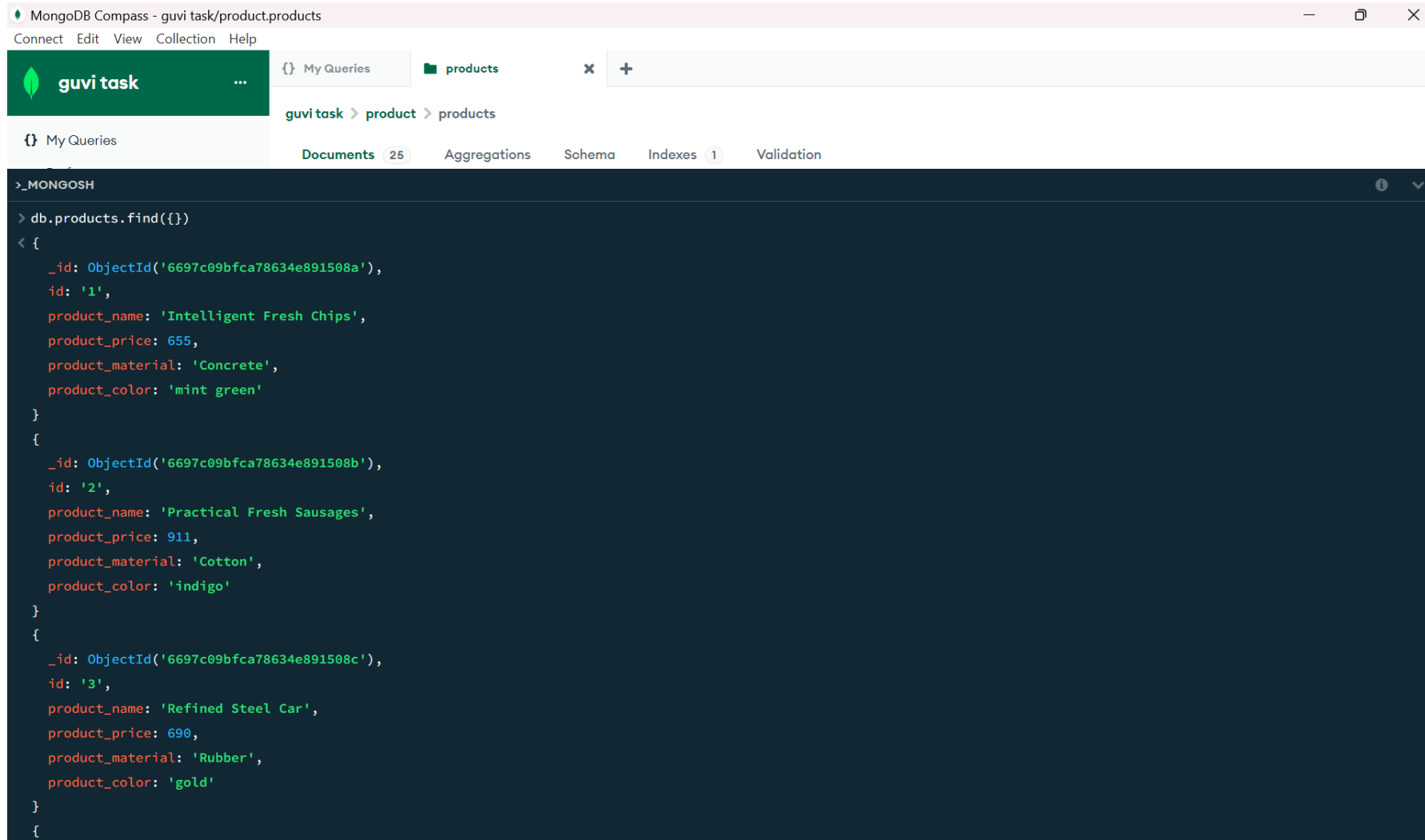
# MongoDB Task Day-1

Name: Getciya S

Email: [getciya7@gmail.com](mailto:getciya7@gmail.com)

## 1. Find all the information about each products

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

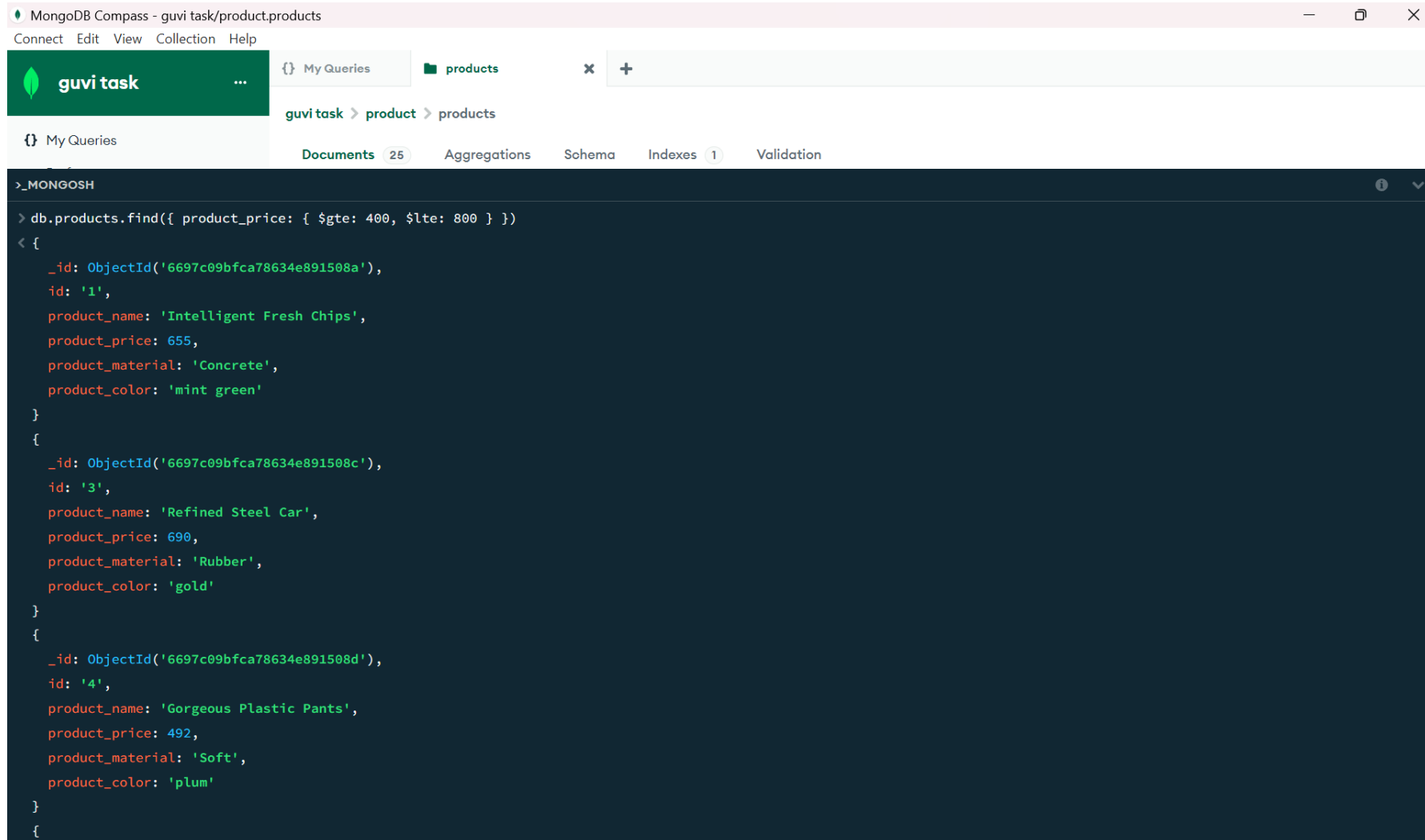


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'guvi task/product.products'. The left sidebar shows the 'guvi task' database and the 'products' collection. The main panel displays the query `db.products.find({})` and its results in a JSON array. The results show three documents with fields: `_id`, `id`, `product_name`, `product_price`, `product_material`, and `product_color`.

```
>_MONGOSH
> db.products.find({})
< [
  {
    _id: ObjectId('6697c09bfca78634e891508a'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508b'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508c'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {
    ...
  }
]
```

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

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

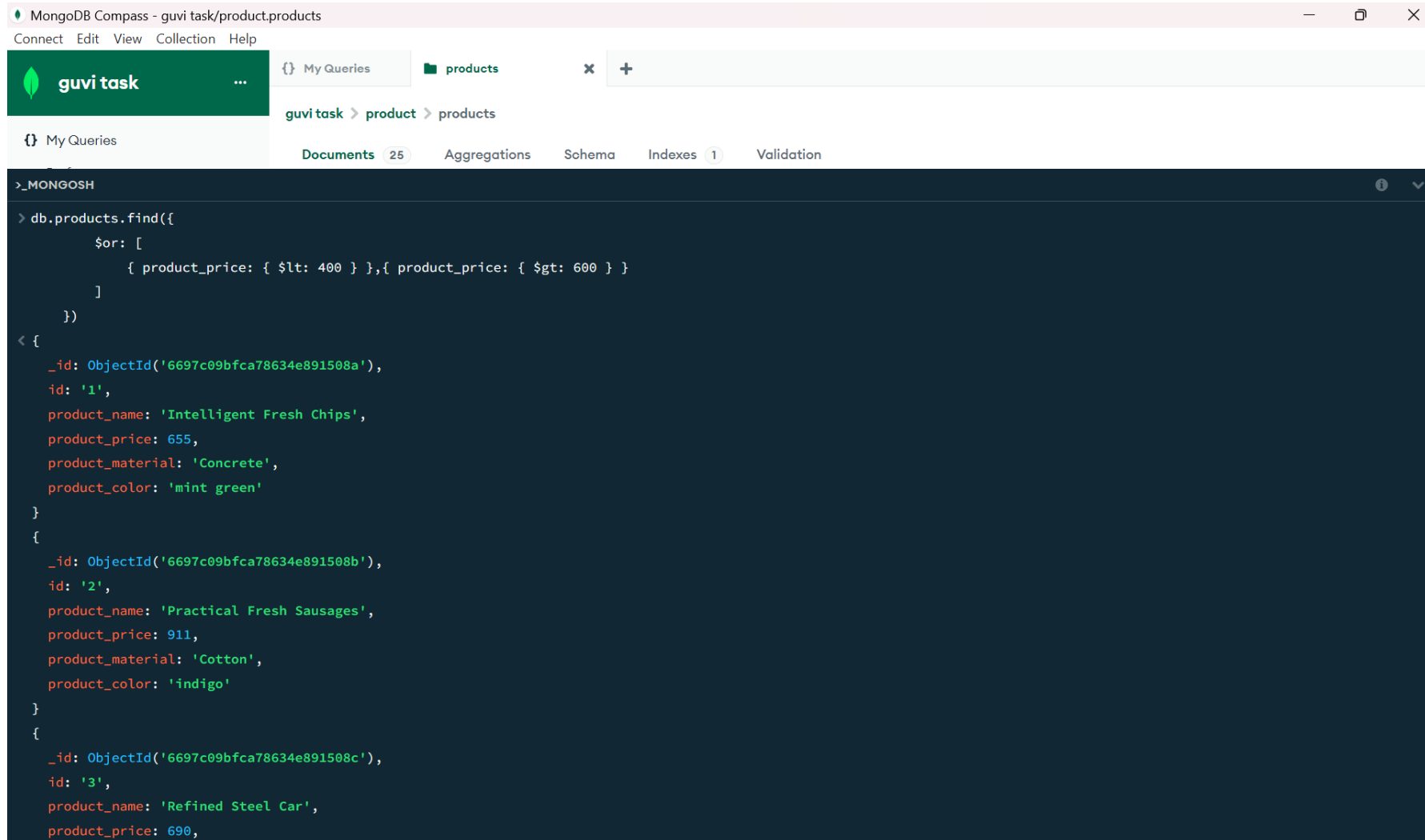


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'guvi task/product.products'. The left sidebar shows the 'guvi task' database and the 'products' collection. The main panel displays the query `db.products.find({ product_price: { $gte: 400, $lte: 800 } })` and its results. The results are shown in a JSON array format, listing three products: 'Intelligent Fresh Chips' (price 655), 'Refined Steel Car' (price 690), and 'Gorgeous Plastic Pants' (price 492). The interface also shows a 'Documents' tab with 25 items and a 'Validation' tab.

```
>_MONGOSH
> db.products.find({ product_price: { $gte: 400, $lte: 800 } })
< [
  {
    _id: ObjectId('6697c09bfca78634e891508a'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508c'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508d'),
    id: '4',
    product_name: 'Gorgeous Plastic Pants',
    product_price: 492,
    product_material: 'Soft',
    product_color: 'plum'
  }
]
```

### 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 } }] })`

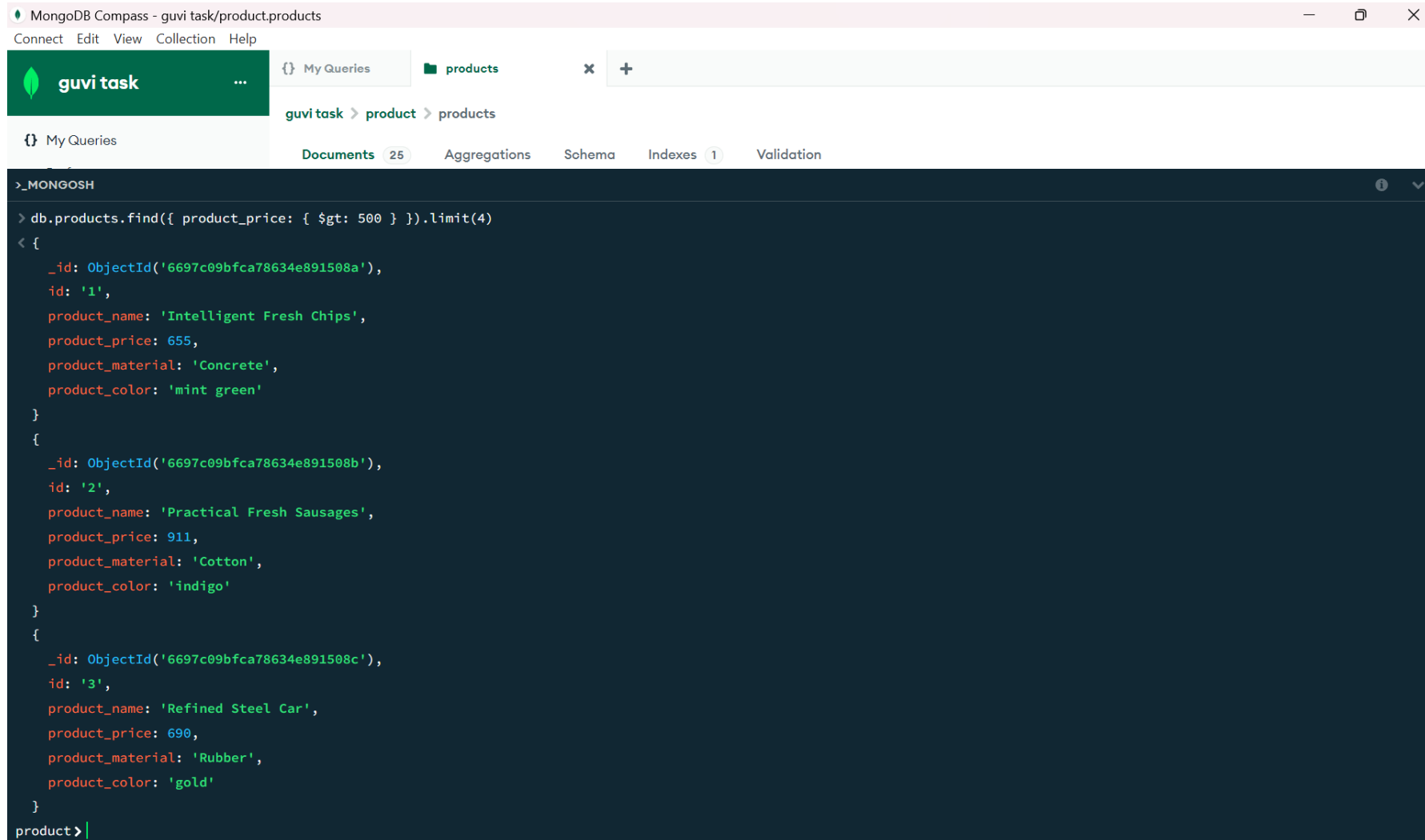


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'guvi task/product.products'. The left sidebar shows the 'guvi task' database and the 'products' collection. The main panel displays the query and its results.

```
>_MONGOSH
> db.products.find({
  $or: [
    { product_price: { $lt: 400 } }, { product_price: { $gt: 600 } }
  ]
})
< {
  _id: ObjectId('6697c09bfca78634e891508a'),
  id: '1',
  product_name: 'Intelligent Fresh Chips',
  product_price: 655,
  product_material: 'Concrete',
  product_color: 'mint green'
}
{
  _id: ObjectId('6697c09bfca78634e891508b'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
}
{
  _id: ObjectId('6697c09bfca78634e891508c'),
  id: '3',
  product_name: 'Refined Steel Car',
  product_price: 690,
```

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

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

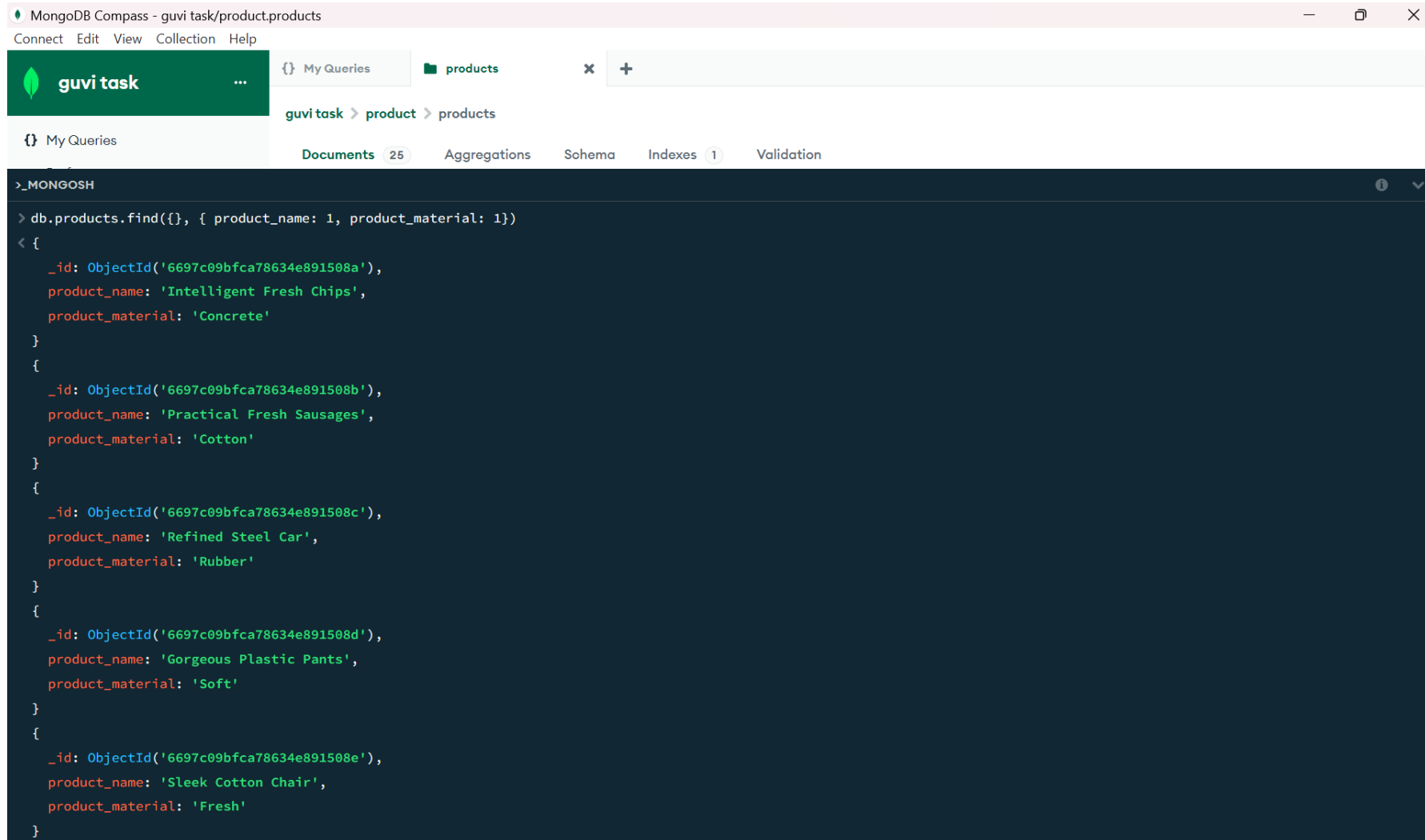


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'guvi task/product.products'. The left sidebar shows the 'guvi task' database and the 'products' collection. The main panel displays the query `db.products.find({ product_price: { $gt: 500 } }).limit(4)` and its results. The results are displayed in a JSON format, showing four products with prices greater than 500.

```
>_MONGOSH
> db.products.find({ product_price: { $gt: 500 } }).limit(4)
< [
  {
    _id: ObjectId('6697c09bfca78634e891508a'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508b'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508c'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 690,
    product_material: 'Rubber',
    product_color: 'gold'
  }
]
```

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

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

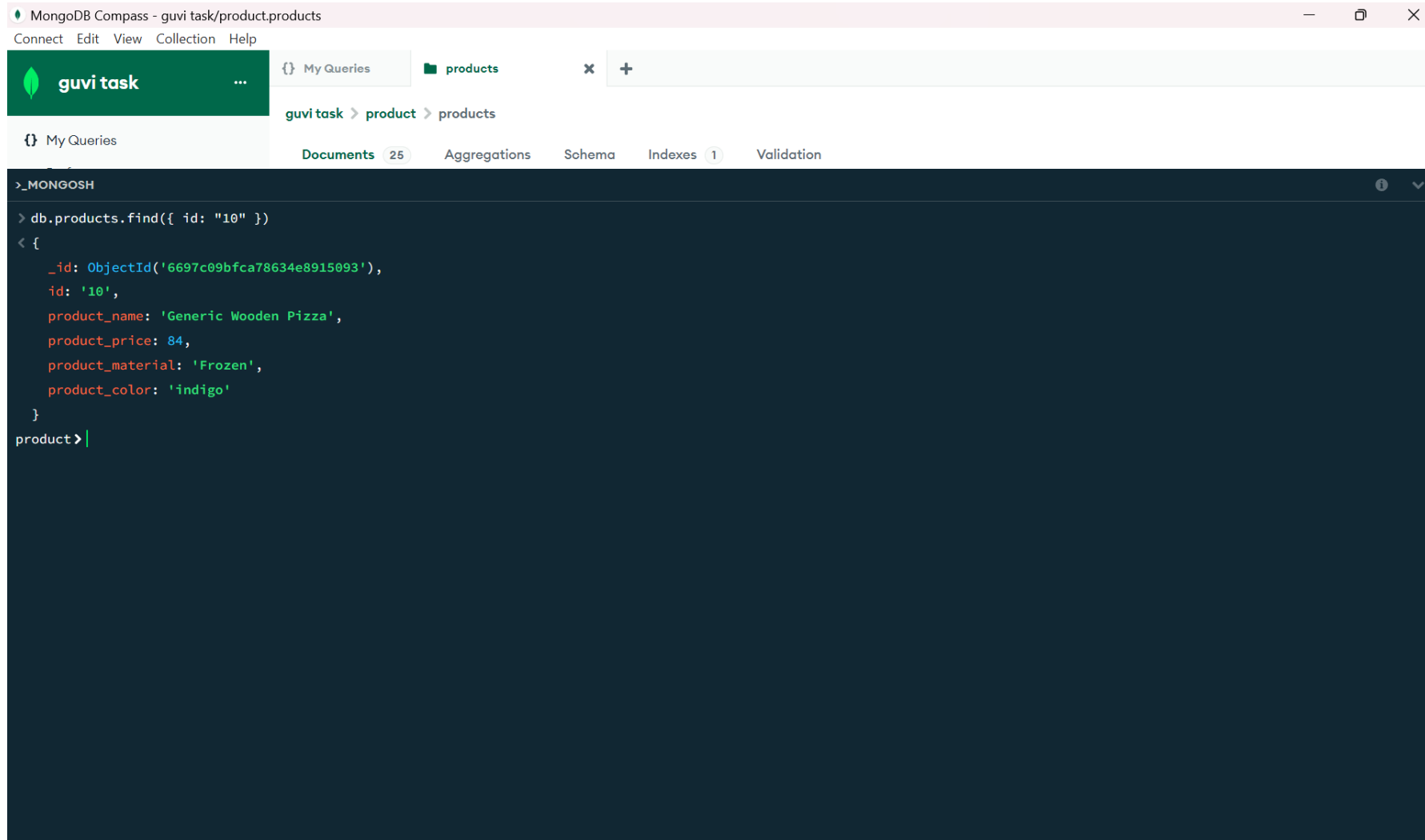


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection is to 'guvi task/product.products'. The left sidebar shows the 'My Queries' tab. The main panel displays the query `db.products.find({}, { product_name: 1, product_material: 1})` and its results. The results are a JSON array of five objects, each containing an ObjectId, a product name, and a product material.

```
>_MONGOSH
> db.products.find({}, { product_name: 1, product_material: 1})
< [
  {
    _id: ObjectId('6697c09bfca78634e891508a'),
    product_name: 'Intelligent Fresh Chips',
    product_material: 'Concrete'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508b'),
    product_name: 'Practical Fresh Sausages',
    product_material: 'Cotton'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508c'),
    product_name: 'Refined Steel Car',
    product_material: 'Rubber'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508d'),
    product_name: 'Gorgeous Plastic Pants',
    product_material: 'Soft'
  },
  {
    _id: ObjectId('6697c09bfca78634e891508e'),
    product_name: 'Sleek Cotton Chair',
    product_material: 'Fresh'
  }
]
```

6. Find the product with a row id of 10

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

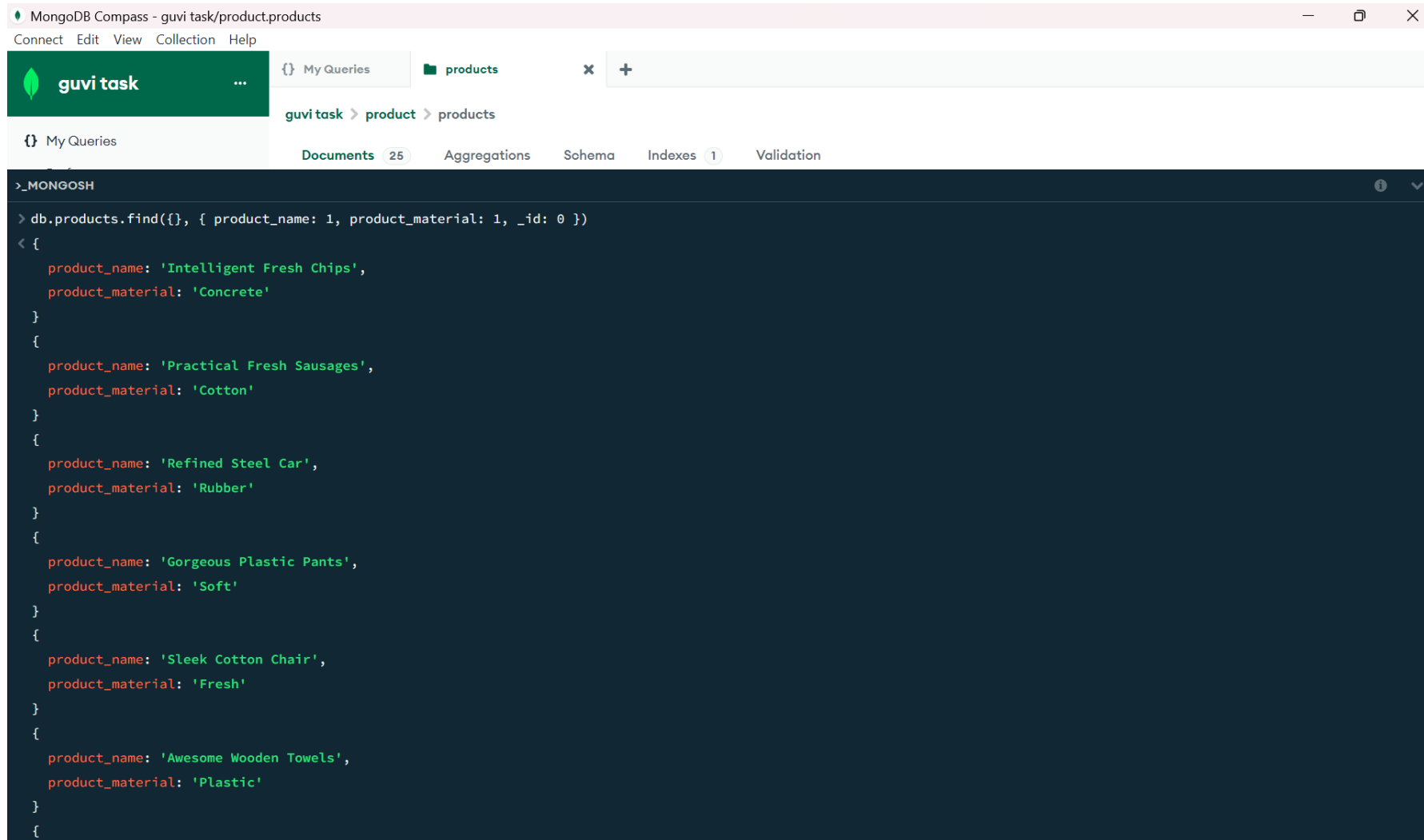


The screenshot shows the MongoDB Compass application window. The title bar reads "MongoDB Compass - guvi task/product.products". The interface includes a top navigation bar with "Connect", "Edit", "View", "Collection", and "Help" menus. Below this is a sidebar with a "guvi task" logo and a "My Queries" tab. The main area displays the "products" collection, with a breadcrumb path "guvi task > product > products". The "Documents" tab is active, showing 25 documents. The query editor at the bottom contains the command `> db.products.find({ id: "10" })`, and the result shows a single document with the following fields: `_id: ObjectId('6697c09bfca78634e8915093')`, `id: '10'`, `product_name: 'Generic Wooden Pizza'`, `product_price: 84`, `product_material: 'Frozen'`, and `product_color: 'indigo'`. The prompt `product>` is visible at the bottom of the query editor.

```
>_MONGOSH
> db.products.find({ id: "10" })
< {
  _id: ObjectId('6697c09bfca78634e8915093'),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 84,
  product_material: 'Frozen',
  product_color: 'indigo'
}
product>
```

## 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 task/product.products'. The left sidebar shows the 'guvi 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 in a JSON array. The results list six products with their names and materials, excluding the \_id field.

```
> 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'
  }
]
```



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

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

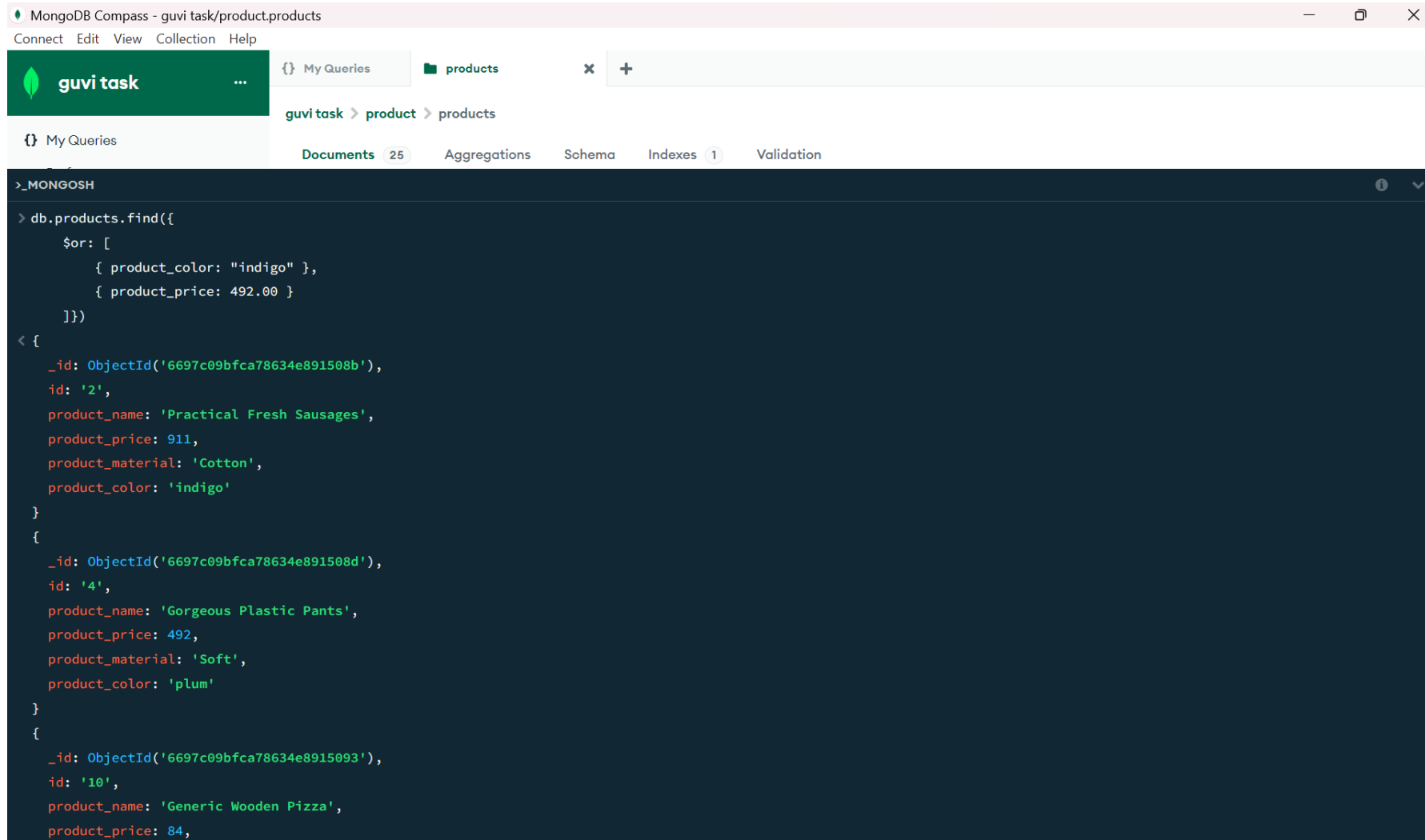
The screenshot displays the MongoDB Compass application interface. At the top, the title bar reads "MongoDB Compass - guvi task/product.products". Below the title bar is a menu bar with "Connect", "Edit", "View", "Collection", and "Help". The main interface is divided into several sections:

- Left Sidebar:** Contains a green header with the MongoDB logo and "guvi task". Below it is a search bar labeled "My Queries".
- Top Bar:** Shows the current collection "products" with a folder icon and a plus sign for adding new collections.
- Breadcrumbs:** Display the path "guvi task > product > products".
- Navigation Tabs:** Includes "Documents" (with a count of 25), "Aggregations", "Schema", "Indexes" (with a count of 1), and "Validation".
- Terminal/Console:** A dark-themed area at the bottom showing a MongoDB shell prompt ">\_MONGOSH". It contains a query and its result:

```
> db.products.find({ product_material: "Soft" })
< {
  _id: ObjectId('6697c09bfca78634e891508d'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
}
{
  _id: ObjectId('6697c09bfca78634e8915092'),
  id: '9',
  product_name: 'Awesome Wooden Ball',
  product_price: 28,
  product_material: 'Soft',
  product_color: 'azure'
}
{
  _id: ObjectId('6697c09bfca78634e8915094'),
  id: '11',
  product_name: 'Unbranded Wooden Cheese',
  product_price: 26,
  product_material: 'Soft',
  product_color: 'black'
}
{
```

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 } ] })`



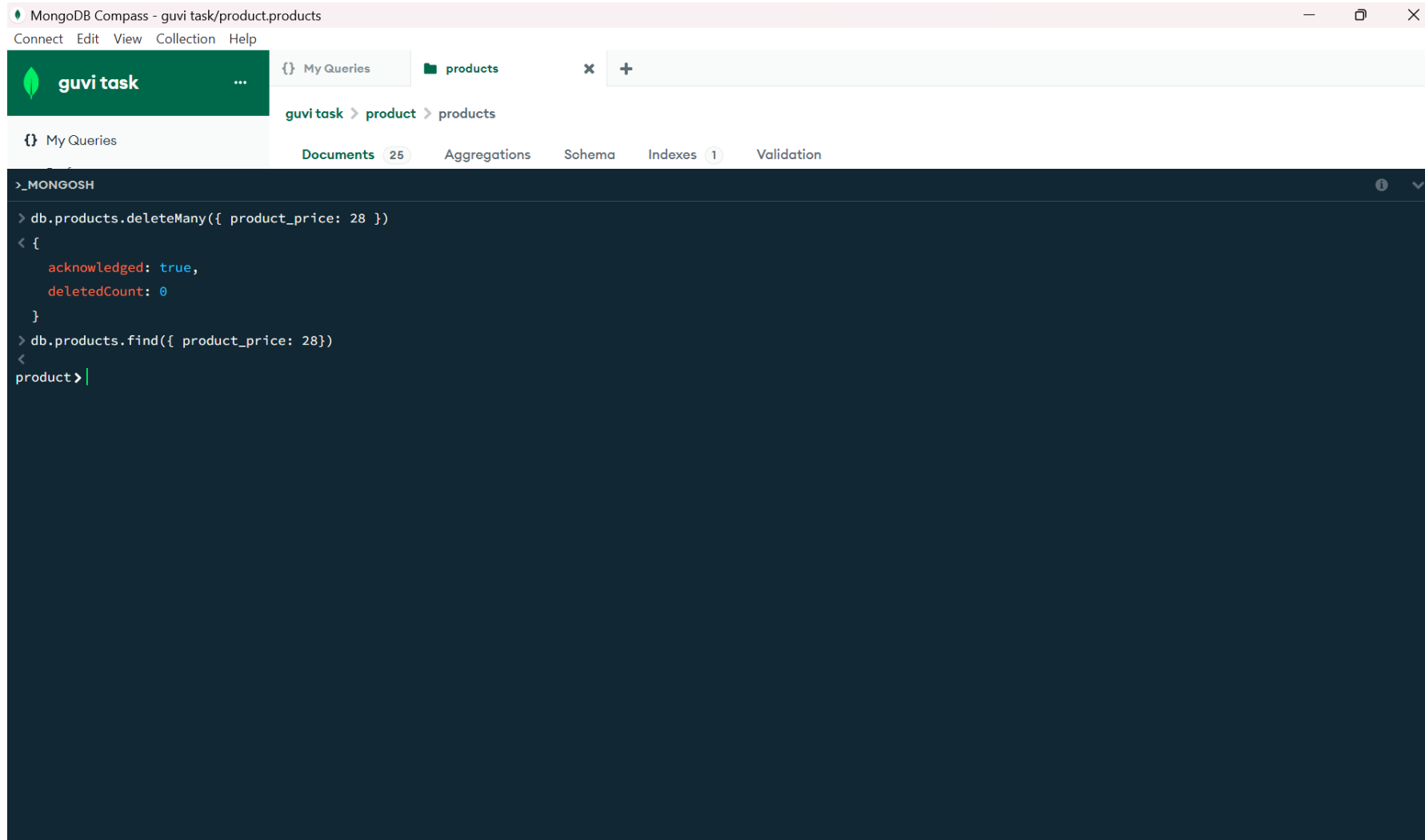
The screenshot shows the MongoDB Compass interface. The top bar indicates the connection is to 'guvi task/product.products'. The left sidebar shows the 'guvi task' database and the 'products' collection. The main panel displays the query and its results.

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

```
< {
  _id: ObjectId('6697c09bfca78634e891508b'),
  id: '2',
  product_name: 'Practical Fresh Sausages',
  product_price: 911,
  product_material: 'Cotton',
  product_color: 'indigo'
}
{
  _id: ObjectId('6697c09bfca78634e891508d'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
}
{
  _id: ObjectId('6697c09bfca78634e8915093'),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 84,
```

10. Delete the products which product price value are 28

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



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection is to 'guvi task/product.products'. The left sidebar shows the 'guvi task' database and the 'products' collection. The main panel displays the 'My Queries' tab with a query editor. The query entered is `db.products.deleteMany({ product_price: 28 })`. Below the query editor, the 'Documents' tab is active, showing 25 documents. The query results are displayed in a dark-themed console window. The output shows the result of the `deleteMany` operation, which is an object with `acknowledged: true` and `deletedCount: 0`. Below this, the `find` operation is shown, which returns an empty array, indicating that no documents were found with the specified price.

```
>_MONGOSH
> db.products.deleteMany({ product_price: 28 })
< {
  acknowledged: true,
  deletedCount: 0
}
> db.products.find({ product_price: 28})
<
product>
```