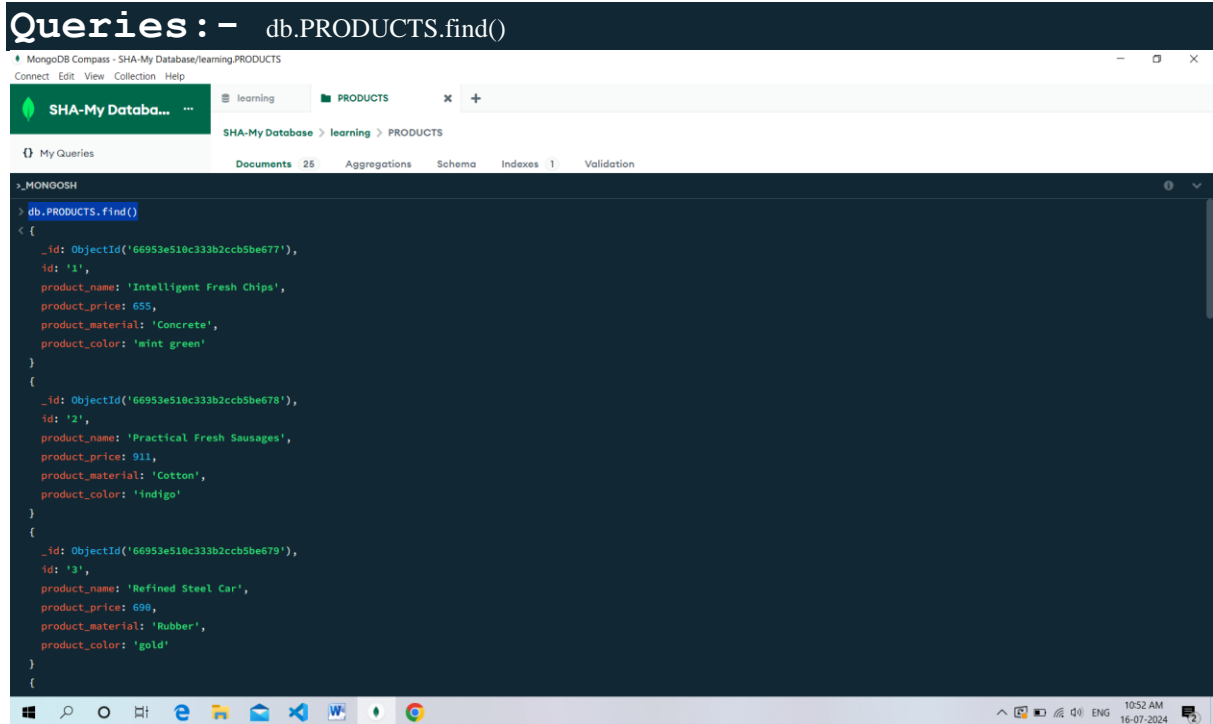


MongoDB Day 1 Task

For the following question write the corresponding MongoDB queries

1. Find all the information about each product

Queries :- `db.PRODUCTS.find()`

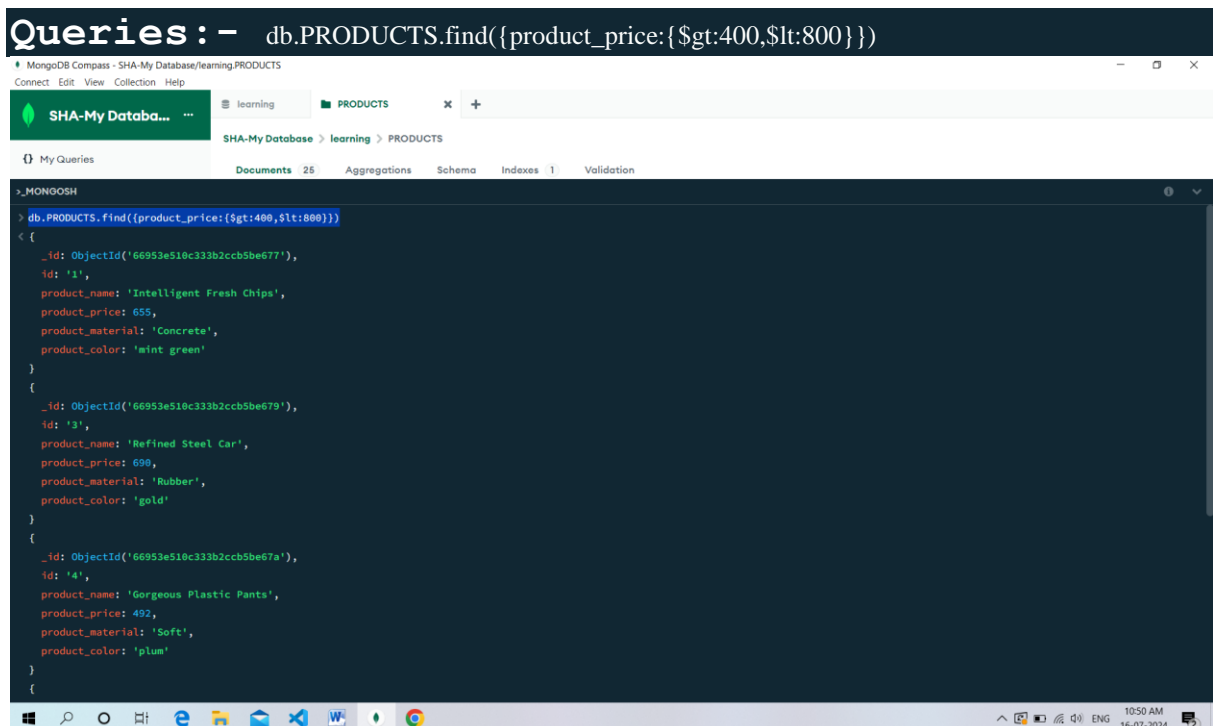


The screenshot shows the MongoDB Compass interface. The top bar indicates the database is 'SHA-My Database' and the collection is 'PRODUCTS'. The 'My Queries' tab is active, showing the query `db.PRODUCTS.find()`. The results are displayed in a JSON array format, showing three products: 'Intelligent Fresh Chips', 'Practical Fresh Sausages', and 'Refined Steel Car'.

```
> db.PRODUCTS.find()
< [
  {
    _id: ObjectId('66953e510c333b2ccb5be677'),
    id: '1',
    product_name: 'Intelligent Fresh Chips',
    product_price: 655,
    product_material: 'Concrete',
    product_color: 'mint green'
  },
  {
    _id: ObjectId('66953e510c333b2ccb5be678'),
    id: '2',
    product_name: 'Practical Fresh Sausages',
    product_price: 911,
    product_material: 'Cotton',
    product_color: 'indigo'
  },
  {
    _id: ObjectId('66953e510c333b2ccb5be679'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 698,
    product_material: 'Rubber',
    product_color: 'gold'
  }
]
```

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

Queries :- `db.PRODUCTS.find({product_price:{$gt:400,$lt:800}})`

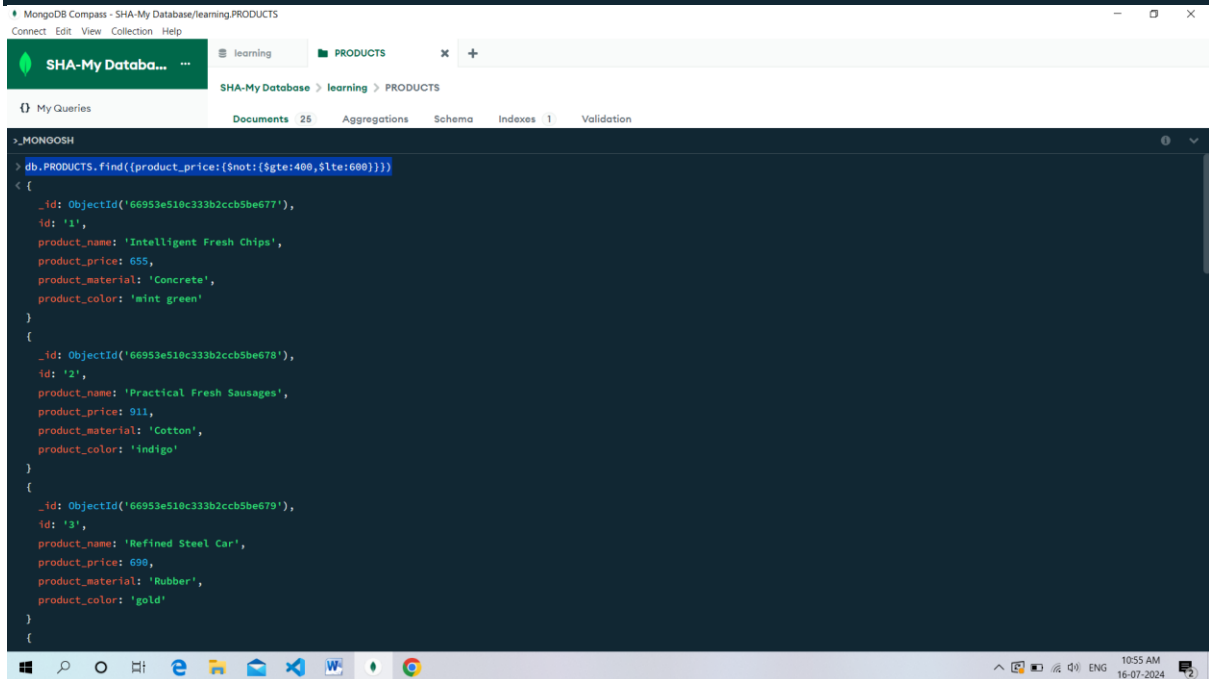


The screenshot shows the MongoDB Compass interface. The top bar indicates the database is 'SHA-My Database' and the collection is 'PRODUCTS'. The 'My Queries' tab is active, showing the query `db.PRODUCTS.find({product_price:{$gt:400,$lt:800}})`. The results are displayed in a JSON array format, showing two products: 'Refined Steel Car' and 'Gorgeous Plastic Pants'.

```
> db.PRODUCTS.find({product_price:{$gt:400,$lt:800}})
< [
  {
    _id: ObjectId('66953e510c333b2ccb5be679'),
    id: '3',
    product_name: 'Refined Steel Car',
    product_price: 698,
    product_material: 'Rubber',
    product_color: 'gold'
  },
  {
    _id: ObjectId('66953e510c333b2ccb5be67a'),
    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

Queries :- `db.PRODUCTS.find({product_price:{$not:{$gte:400,$lte:600}}})`

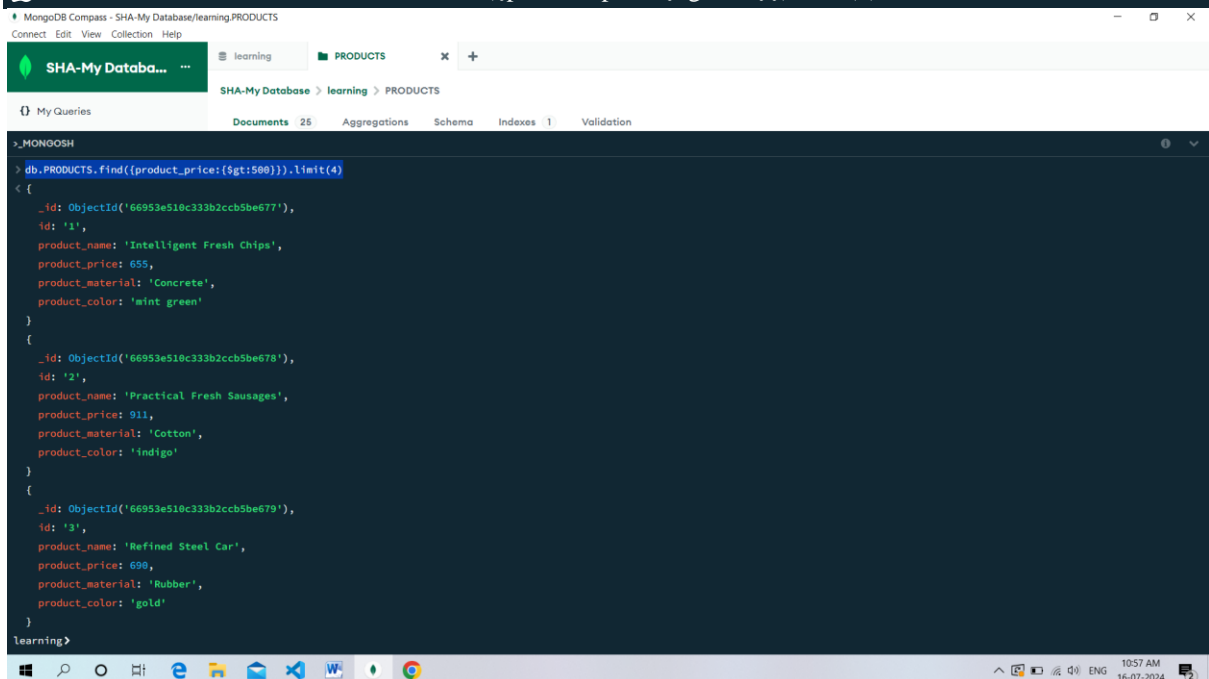


The screenshot shows the MongoDB Compass interface. The query `db.PRODUCTS.find({product_price:{$not:{$gte:400,$lte:600}}})` is entered in the command bar. The results pane shows three documents:

```
{
  "_id": ObjectId("66953e510c333b2ccb5be677"),
  "id": "1",
  "product_name": "Intelligent Fresh Chips",
  "product_price": 655,
  "product_material": "Concrete",
  "product_color": "mint green"
},
{
  "_id": ObjectId("66953e510c333b2ccb5be678"),
  "id": "2",
  "product_name": "Practical Fresh Sausages",
  "product_price": 911,
  "product_material": "Cotton",
  "product_color": "indigo"
},
{
  "_id": ObjectId("66953e510c333b2ccb5be679"),
  "id": "3",
  "product_name": "Refined Steel Car",
  "product_price": 698,
  "product_material": "Rubber",
  "product_color": "gold"
}
```

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

Queries :- `db.PRODUCTS.find({product_price:{$gt:500}}).limit(4)`

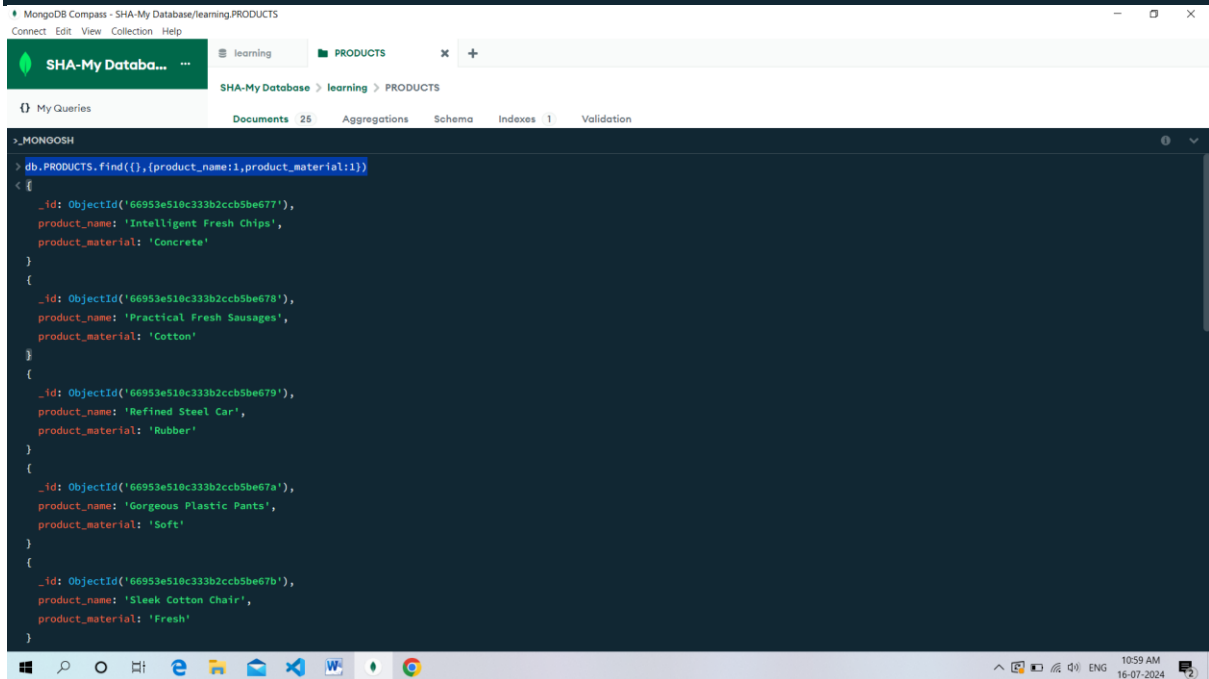


The screenshot shows the MongoDB Compass interface. The query `db.PRODUCTS.find({product_price:{$gt:500}}).limit(4)` is entered in the command bar. The results pane shows three documents (the fourth is truncated):

```
{
  "_id": ObjectId("66953e510c333b2ccb5be677"),
  "id": "1",
  "product_name": "Intelligent Fresh Chips",
  "product_price": 655,
  "product_material": "Concrete",
  "product_color": "mint green"
},
{
  "_id": ObjectId("66953e510c333b2ccb5be678"),
  "id": "2",
  "product_name": "Practical Fresh Sausages",
  "product_price": 911,
  "product_material": "Cotton",
  "product_color": "indigo"
},
{
  "_id": ObjectId("66953e510c333b2ccb5be679"),
  "id": "3",
  "product_name": "Refined Steel Car",
  "product_price": 698,
  "product_material": "Rubber",
  "product_color": "gold"
}
```

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

Queries :- `db.PRODUCTS.find({}, {product_name:1, product_material:1})`

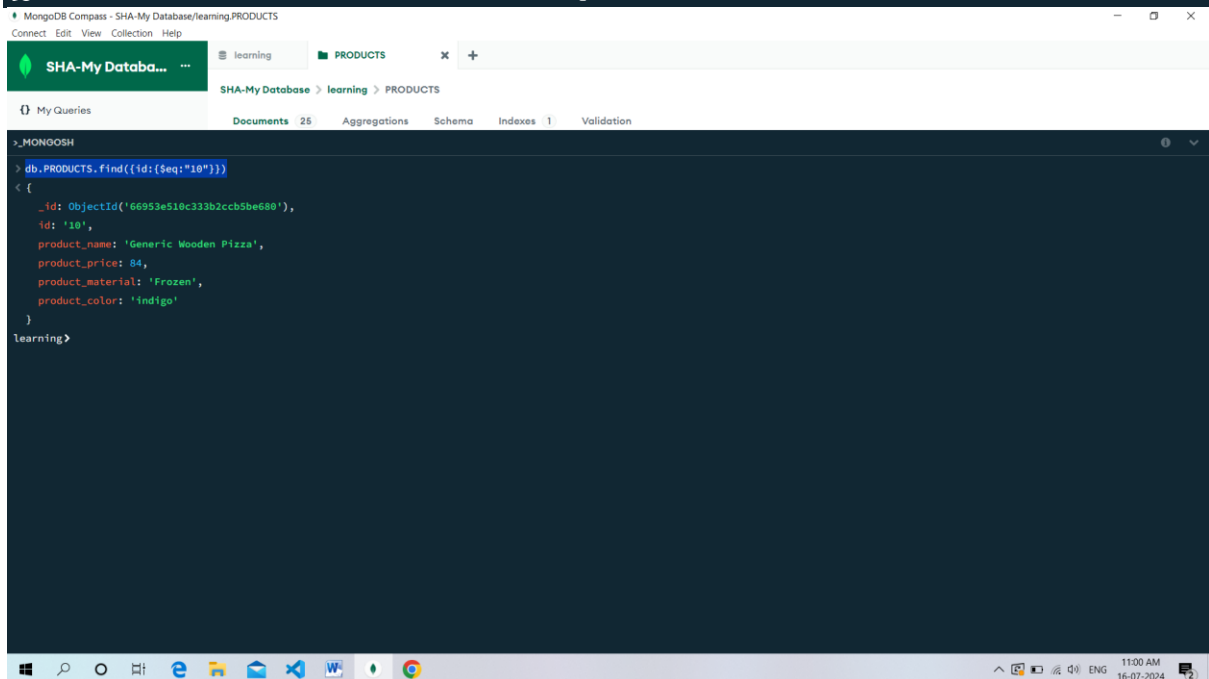


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'SHA-My Database/learning/PRODUCTS'. The left sidebar shows the 'My Queries' tab. The main area displays the query `db.PRODUCTS.find({}, {product_name:1, product_material:1})` and its results. The results are a list of five documents, each containing the product name and material.

```
>_MONGOOSH
> db.PRODUCTS.find({}, {product_name:1, product_material:1})
< {
  _id: ObjectId('66953e510c33b2ccb5be677'),
  product_name: 'Intelligent Fresh Chips',
  product_material: 'Concrete'
}
{
  _id: ObjectId('66953e510c33b2ccb5be678'),
  product_name: 'Practical Fresh Sausages',
  product_material: 'Cotton'
}
{
  _id: ObjectId('66953e510c33b2ccb5be679'),
  product_name: 'Refined Steel Car',
  product_material: 'Rubber'
}
{
  _id: ObjectId('66953e510c33b2ccb5be67a'),
  product_name: 'Gorgeous Plastic Pants',
  product_material: 'Soft'
}
{
  _id: ObjectId('66953e510c33b2ccb5be67b'),
  product_name: 'Sleek Cotton Chair',
  product_material: 'Fresh'
}
```

6. Find the product with a row id of 10

Queries :- `db.PRODUCTS.find({id:{$eq:"10"}})`

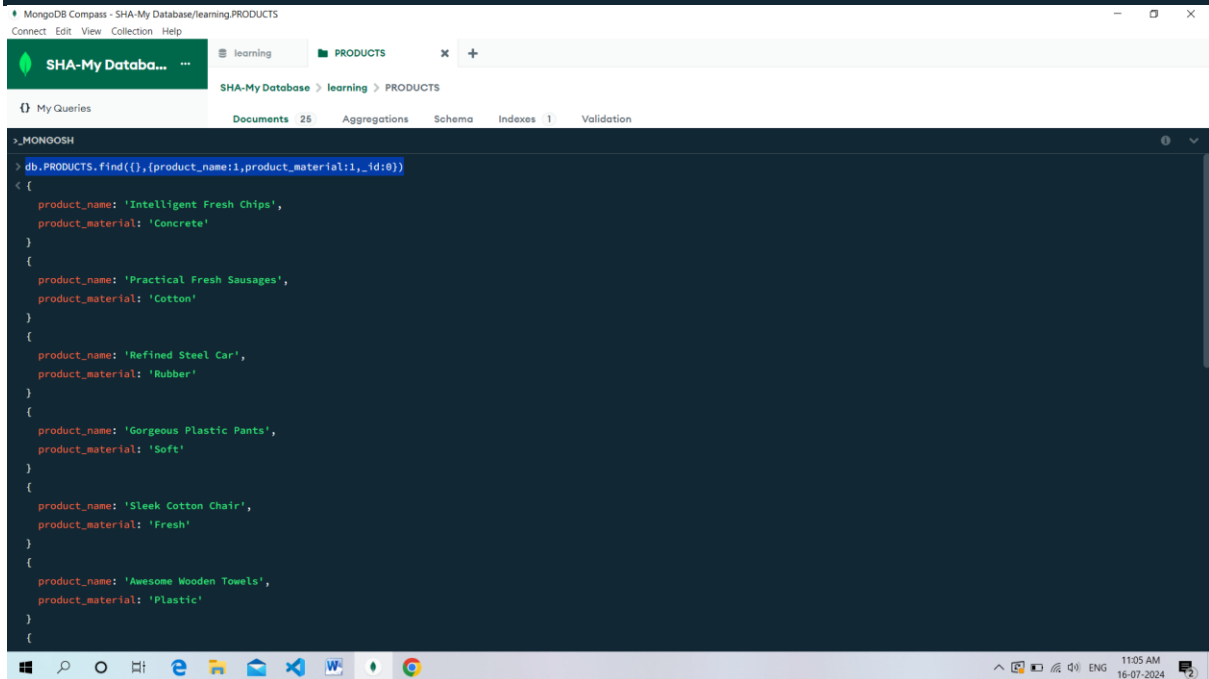


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'SHA-My Database/learning/PRODUCTS'. The left sidebar shows the 'My Queries' tab. The main area displays the query `db.PRODUCTS.find({id:{$eq:"10"}})` and its results. The results are a single document containing the product name, price, material, and color.

```
>_MONGOOSH
> db.PRODUCTS.find({id:{$eq:"10"}})
< {
  _id: ObjectId('66953e510c33b2ccb5be688'),
  id: '10',
  product_name: 'Generic Wooden Pizza',
  product_price: 84,
  product_material: 'Frozen',
  product_color: 'Indigo'
}
learning>
```

7. Find only the product name and product material

Queries :- `db.PRODUCTS.find({}, {product_name:1, product_material:1, _id:0})`

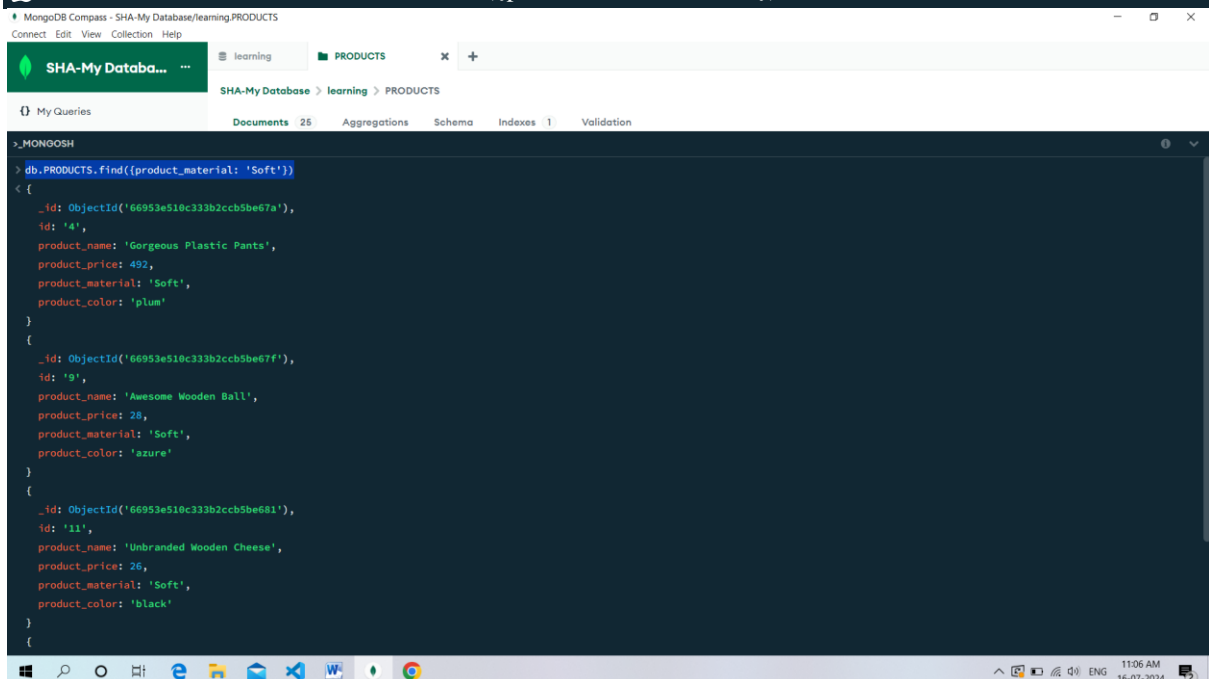


The screenshot shows the MongoDB Compass interface. The top bar indicates the database is 'SHA-My Database' and the collection is 'PRODUCTS'. The query editor shows the query: `db.PRODUCTS.find({}, {product_name:1, product_material:1, _id:0})`. The results pane displays a list of products with their names and materials.

```
>_MONGODB
> 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

Queries :- `db.PRODUCTS.find({product_material: 'Soft'})`

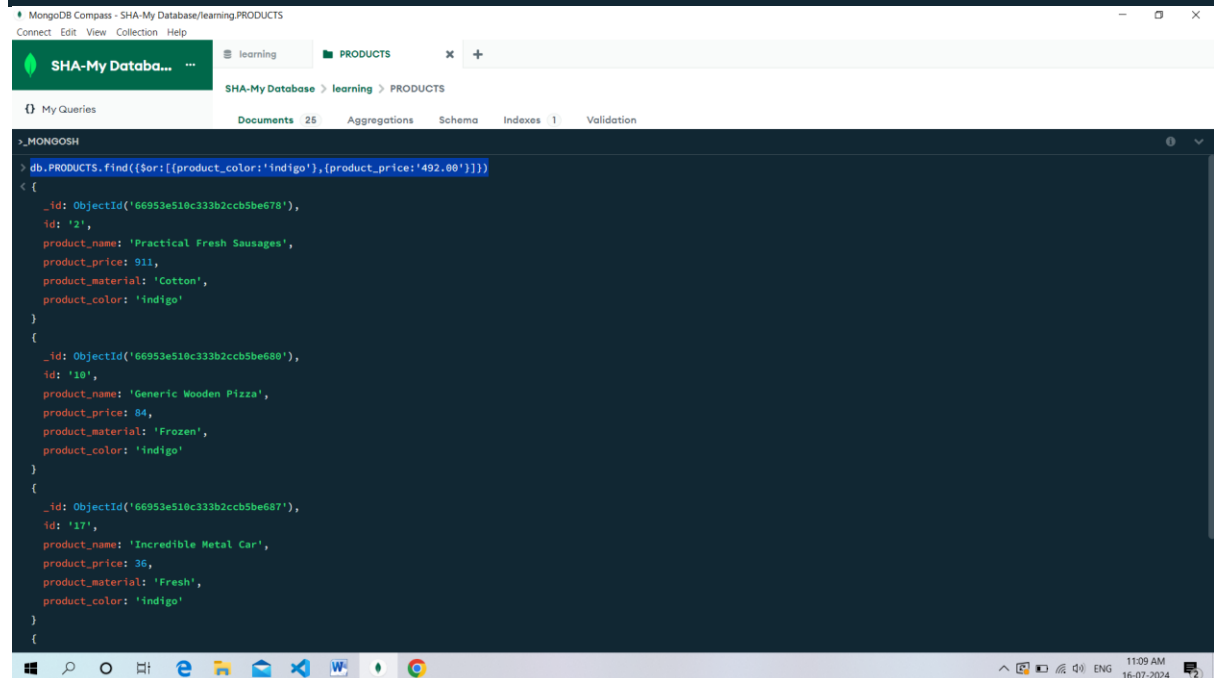


The screenshot shows the MongoDB Compass interface. The top bar indicates the database is 'SHA-My Database' and the collection is 'PRODUCTS'. The query editor shows the query: `db.PRODUCTS.find({product_material: 'Soft'})`. The results pane displays a list of products with their names, materials, and other details.

```
>_MONGODB
> db.PRODUCTS.find({product_material: 'Soft'})
< {
  _id: ObjectId('66953e510c33b2ccb5be67a'),
  id: '4',
  product_name: 'Gorgeous Plastic Pants',
  product_price: 492,
  product_material: 'Soft',
  product_color: 'plum'
}
{
  _id: ObjectId('66953e510c33b2ccb5be67f'),
  id: '9',
  product_name: 'Awesome Wooden Ball',
  product_price: 28,
  product_material: 'Soft',
  product_color: 'azure'
}
{
  _id: ObjectId('66953e510c33b2ccb5be681'),
  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

Queries :- `db.PRODUCTS.find({$or:[{product_color:'indigo'},{product_price:'492.00'}]})`

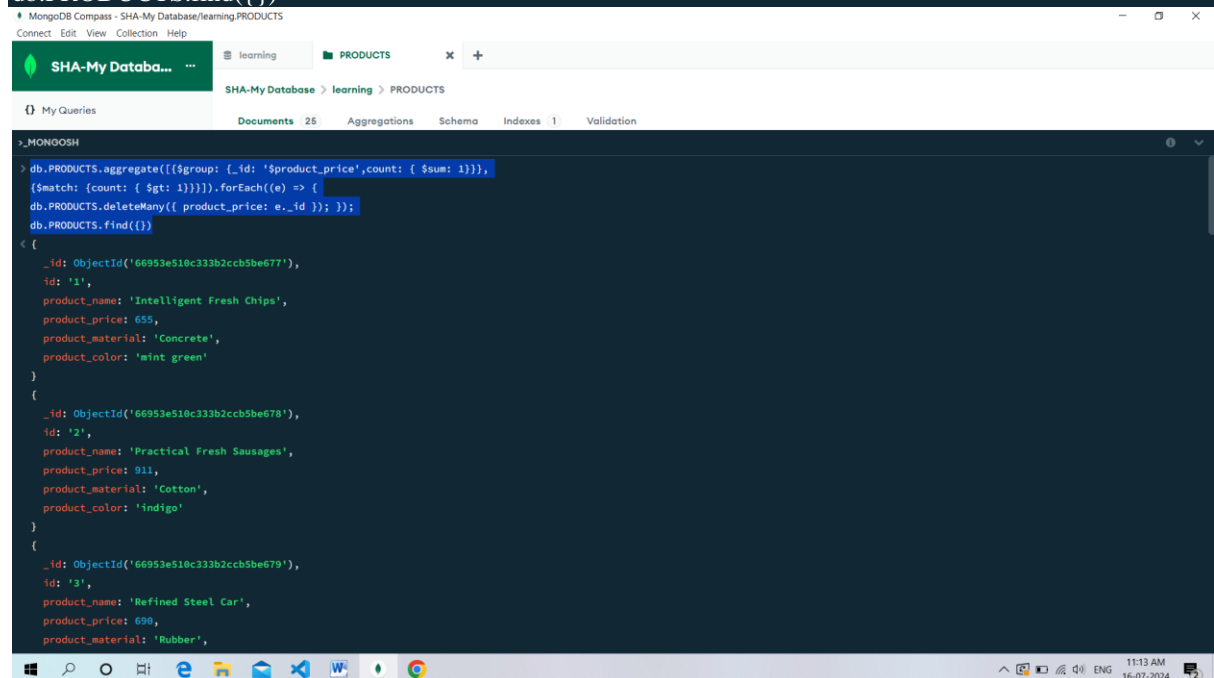


The screenshot shows the MongoDB Compass interface. The query `db.PRODUCTS.find({$or:[{product_color:'indigo'},{product_price:'492.00'}]})` is entered in the command bar. The results pane shows three documents:

```
{
  "_id": ObjectId("66953e510c33b2ccb5be678"),
  "id": "2",
  "product_name": "Practical Fresh Sausages",
  "product_price": 911,
  "product_material": "Cotton",
  "product_color": "indigo"
},
{
  "_id": ObjectId("66953e510c33b2ccb5be688"),
  "id": "10",
  "product_name": "Generic Wooden Pizza",
  "product_price": 84,
  "product_material": "Frozen",
  "product_color": "indigo"
},
{
  "_id": ObjectId("66953e510c33b2ccb5be687"),
  "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 same

Queries :- `db.PRODUCTS.aggregate([{$group: {_id: '$product_price', count: { $sum: 1 } }}, {$match: {count: { $gt: 1 }}}]).forEach((e) => { db.PRODUCTS.deleteMany({ product_price: e._id }); }); db.PRODUCTS.find({})`



The screenshot shows the MongoDB Compass interface. The aggregation query `db.PRODUCTS.aggregate([{$group: {_id: '$product_price', count: { $sum: 1 } }}, {$match: {count: { $gt: 1 }}}]).forEach((e) => { db.PRODUCTS.deleteMany({ product_price: e._id }); }); db.PRODUCTS.find({})` is entered in the command bar. The results pane shows three documents:

```
{
  "_id": ObjectId("66953e510c33b2ccb5be677"),
  "id": "1",
  "product_name": "Intelligent Fresh Chips",
  "product_price": 655,
  "product_material": "Concrete",
  "product_color": "mint green"
},
{
  "_id": ObjectId("66953e510c33b2ccb5be678"),
  "id": "2",
  "product_name": "Practical Fresh Sausages",
  "product_price": 911,
  "product_material": "Cotton",
  "product_color": "indigo"
},
{
  "_id": ObjectId("66953e510c33b2ccb5be679"),
  "id": "3",
  "product_name": "Refined Steel Car",
  "product_price": 698,
  "product_material": "Rubber",
  "product_color": "indigo"
}
```