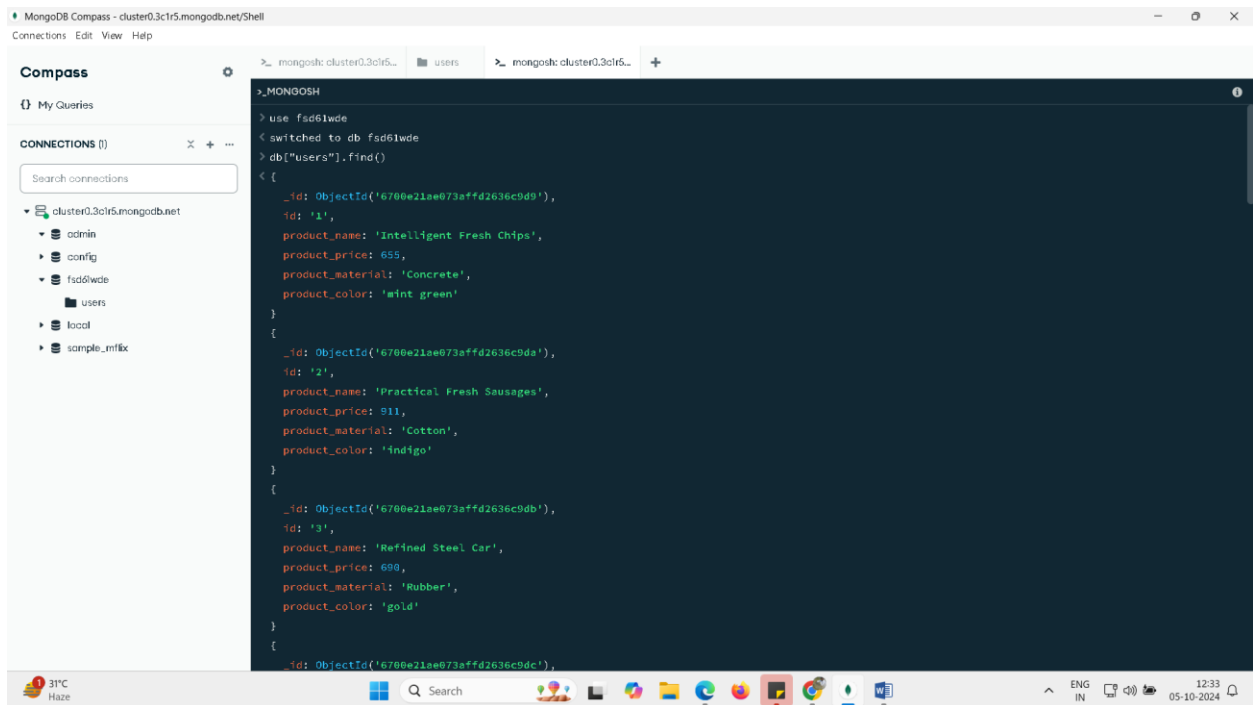


For the following question write the corresponding MongoDB queries

1. Find all the information about each product

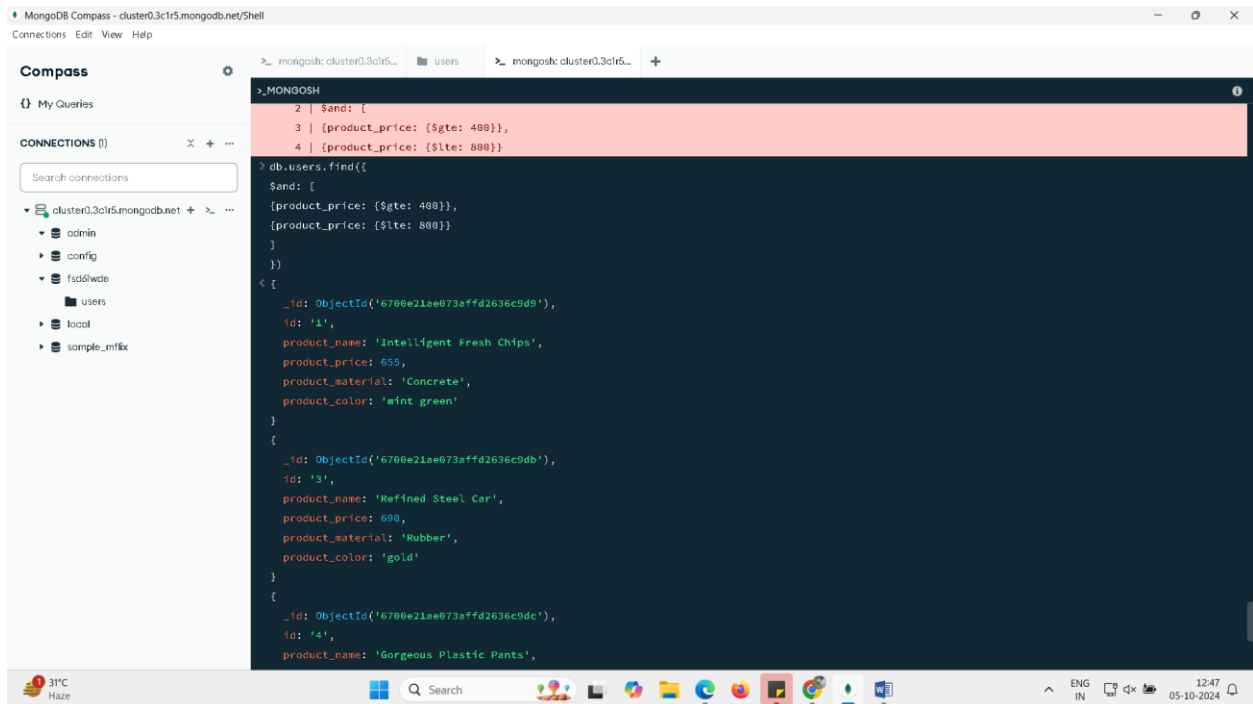
Answer: `Db.users.find()`



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

Answer:

```
db.users.find({
  $and: [
    {product_price: {$gte: 400}},
    {product_price: {$lte: 800}}
  ]
})
```



The screenshot shows the MongoDB Compass interface. The left sidebar displays the 'Connections' list with 'cluster0.3c1r5.mongodb.net' selected. The main panel shows a query in the 'MongoShell' tab. The query is a find operation on the 'users' collection using an \$and operator to filter by product price. The results are displayed in a table with three rows of user data.

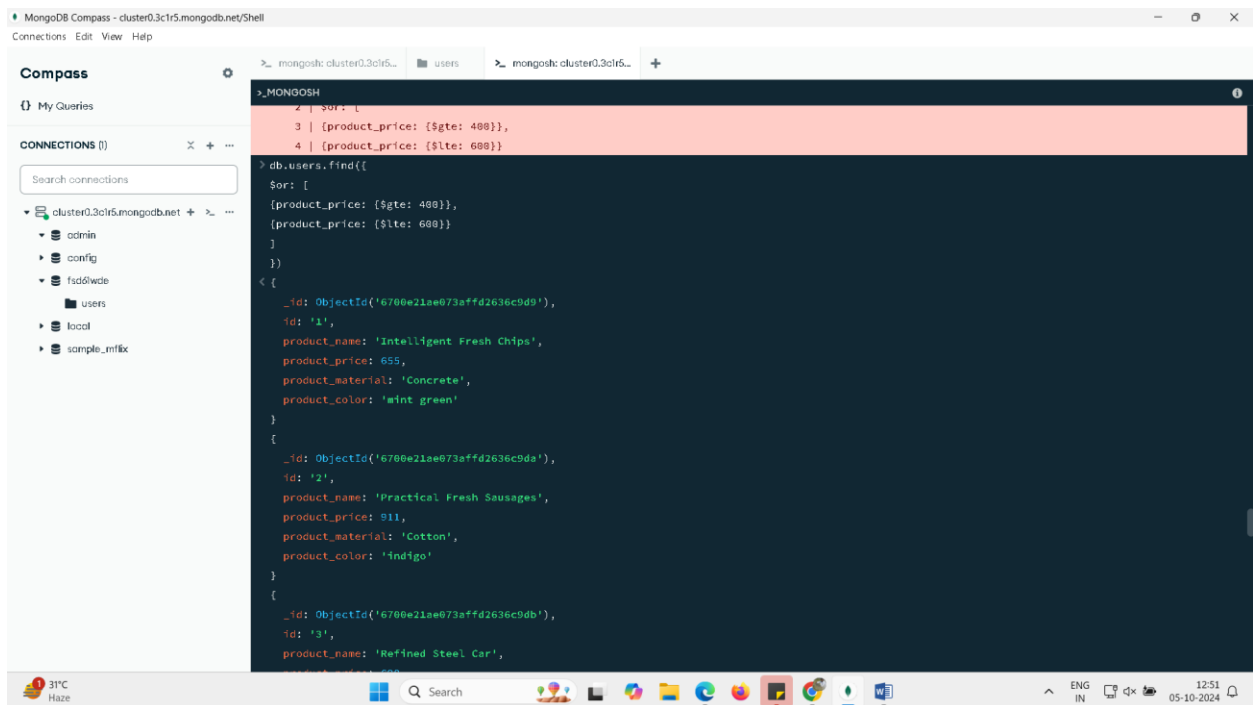
```
> use users
> db.users.find(
  2 | $and: [
  3 |   {product_price: {$gte: 400}},
  4 |   {product_price: {$lte: 800}}
  )
{
  "_id" : ObjectId("6706e21ae073affd2636c9d9"),
  "id" : "1",
  "product_name" : "Intelligent Fresh Chips",
  "product_price" : 655,
  "product_material" : "Concrete",
  "product_color" : "mint green"
}
{
  "_id" : ObjectId("6706e21ae073affd2636c9db"),
  "id" : "3",
  "product_name" : "Refined Steel Car",
  "product_price" : 698,
  "product_material" : "Rubber",
  "product_color" : "gold"
}
{
  "_id" : ObjectId("6706e21ae073affd2636c9dc"),
  "id" : "4",
  "product_name" : "Gorgeous Plastic Pants",
```

id	product_name	product_price	product_material	product_color
1	Intelligent Fresh Chips	655	Concrete	mint green
3	Refined Steel Car	698	Rubber	gold
4	Gorgeous Plastic Pants			

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

Answer:

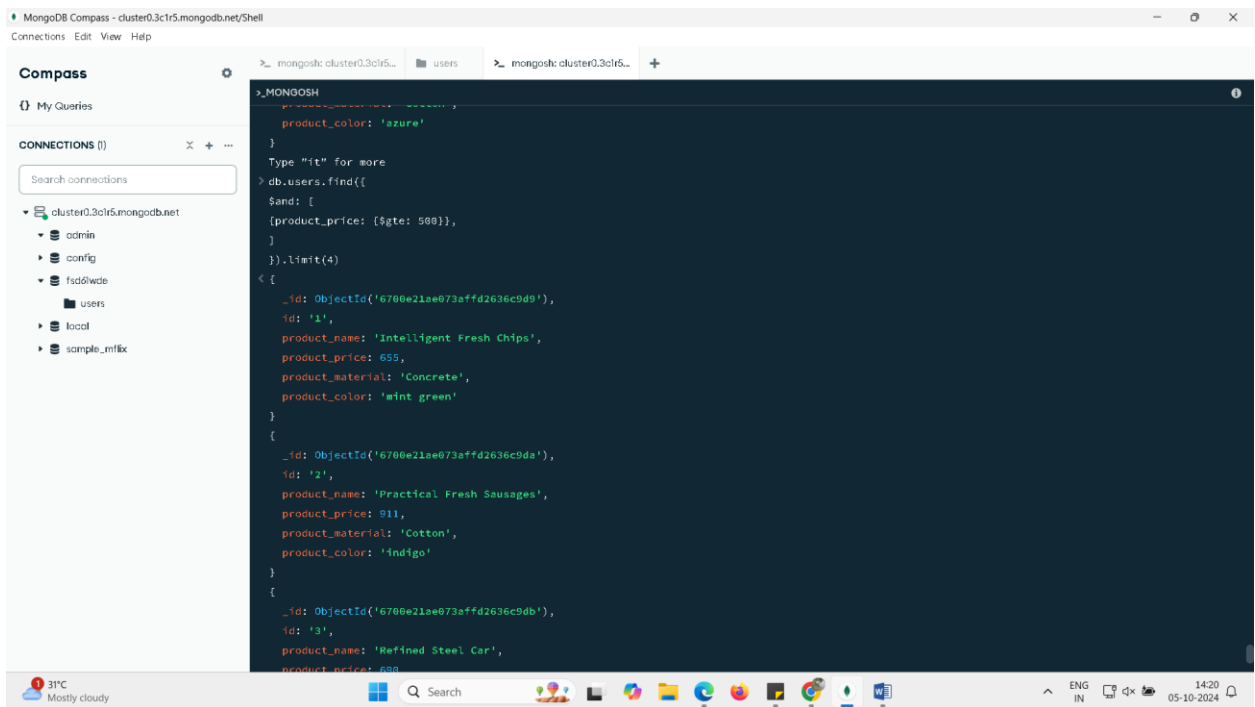
```
db.users.find({
  $or: [
    {product_price: {$gte: 400}},
    {product_price: {$lte: 600}}
  ]
})
```



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

Answer:

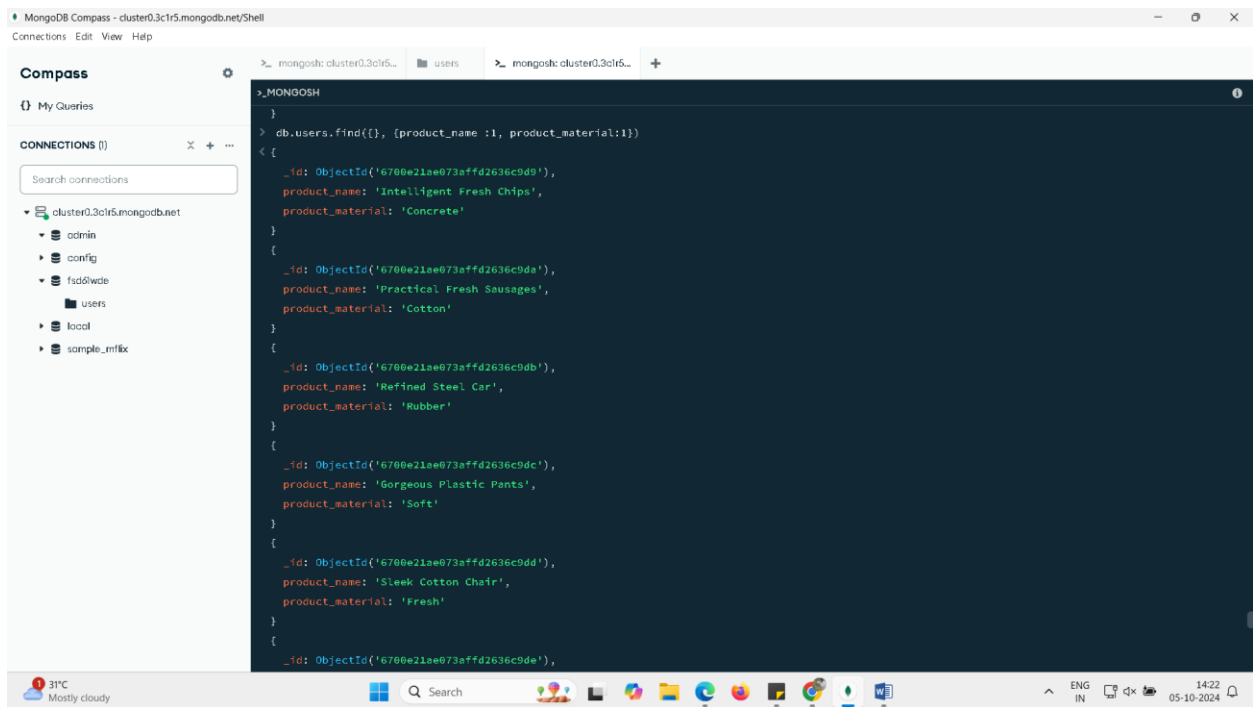
```
db.users.find({
  $and: [
    {product_price: {$gte: 500}},
  ]
}).limit(4)
```



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

Answer:

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



6. Find the product with a row id of 10

Answer:

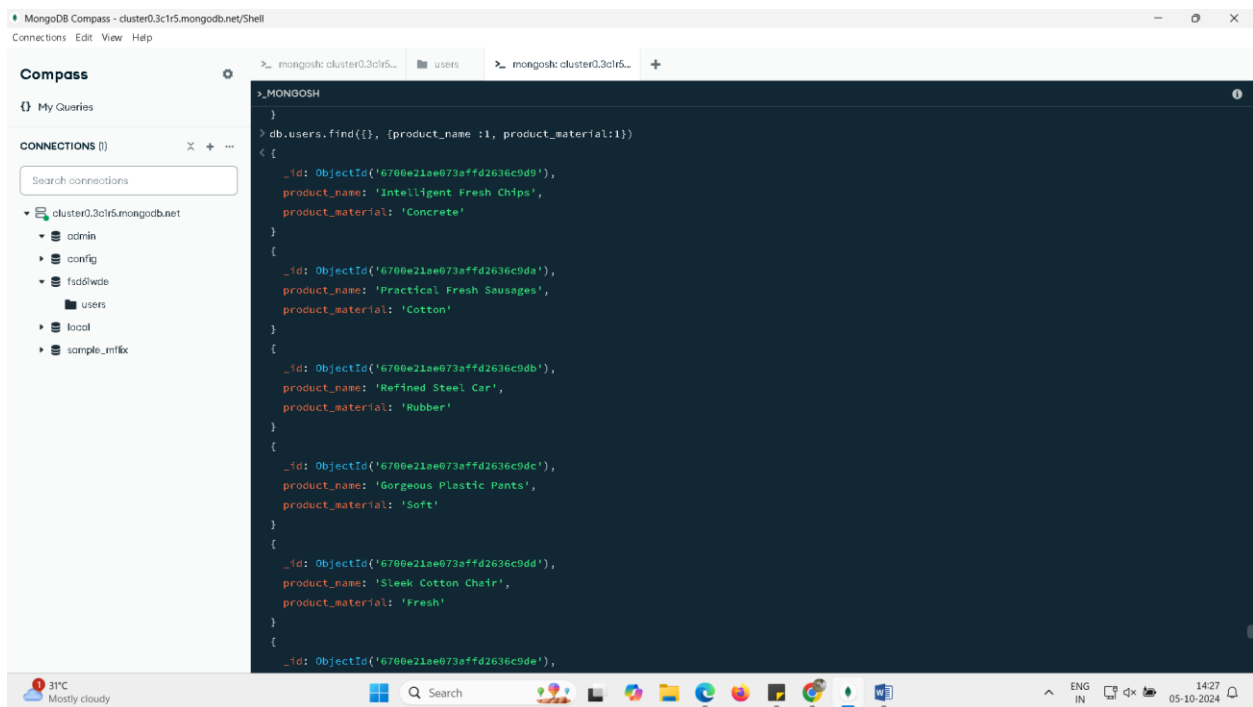
```
db.products.find({ id:10})
```

```
> db.products.find({ id:'10'})
< [
  {
    _id: ObjectId('6700e21ae073affd2636c9e2'),
    id: '10',
    product_name: 'Generic Wooden Pizza',
    product_price: 84,
    product_material: 'Frozen',
    product_color: 'Indigo'
  }
]
Atlas atlas-4kojwa-shard-0 [primary] fsd6lwd>
```

7. Find only the product name and product material

Answer:

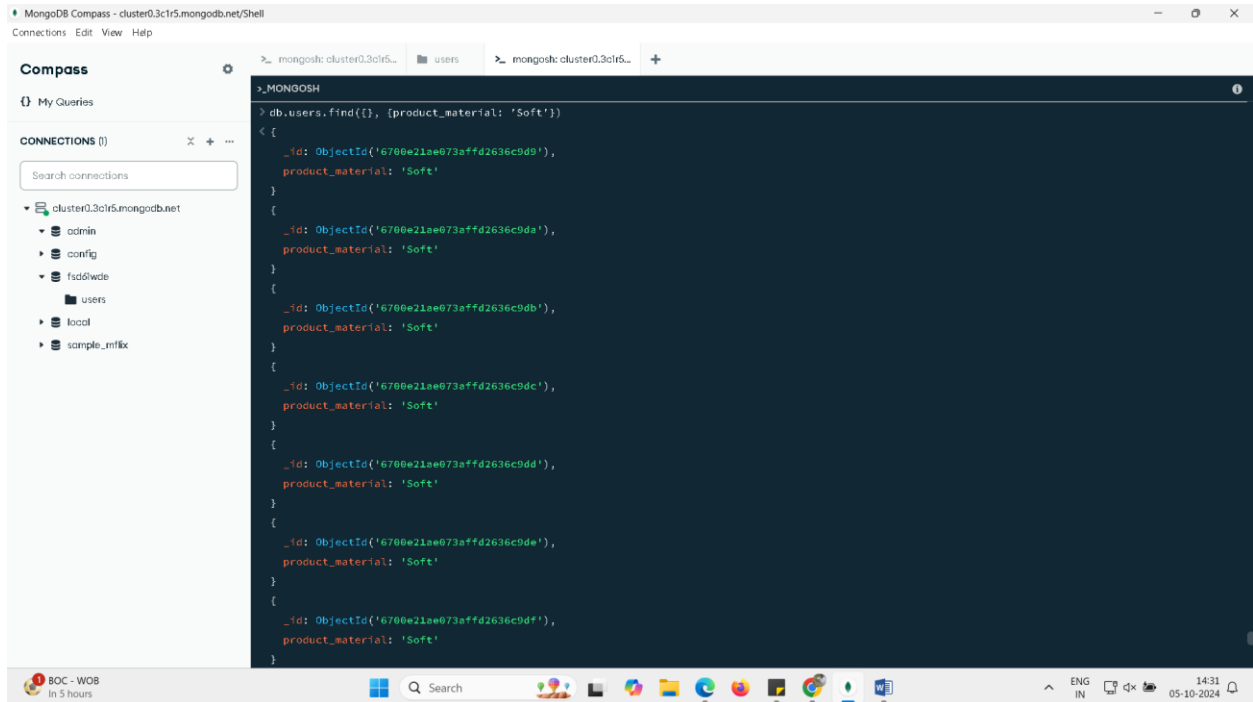
```
db.products.find({}, {product_name :1, product_material:1})
```



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

Answer:

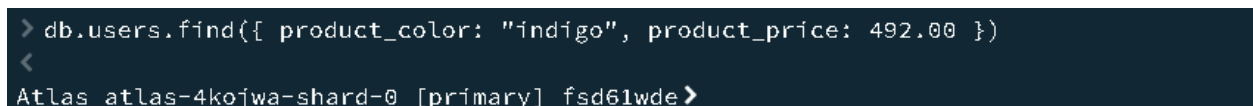
```
db.users.find({}, {product_material:'Soft'})
```



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

Answer:

```
db.users.find( {},{product_price: '492', product_color: 'indigo'})
```



10. Delete the products which product price value are 28

Answer:

```
db.users.deleteMany({ product_price: 28 })
```

