Experiment – 6: MongoDB

Name of the Student	**Ria Chaudhari**
Class and Roll No	**D15A07**
DOP	
DOS	
Sign and Grade	

Aim: To study CRUD operations in MongoDB

Problem Statement:

Create a database, create a collection, insert data, query and manipulate data using various MongoDB operations.

1.Create a database named "inventory".
2.Create a collection named "products" with the fields: (ProductID, ProductName, Category, Price, Stock).
3.Insert 10 documents into the "products" collection.
4.Display all the documents in the "products" collection.
5.Display all the products in the "Electronics" category.
6.Display all the products in ascending order of their names.
7.Display the details of the first 5 products.
8.Display the categories of products with a specific name.
9.Display the number of products in the "Electronics" category.
10.Display all the products without showing the "_id" field.
11.Display all the distinct categories of products.
12.Display products in the "Electronics" category with prices greater than 50 but less than 100.
13.Change the price of a product.
14.Delete a particular product entry.

Theory:

A)Describe some of the features of MongoDB?

Features of MongoDB

MongoDB is a NoSQL database known for its scalability, flexibility, and high performance. Some key features include:

Schema-less Database – Unlike relational databases, MongoDB does not require a fixed schema, allowing dynamic and flexible data storage.
Document-Oriented Storage – Data is stored in BSON (Binary JSON) format, making it easy to work with hierarchical data.

Scalability – MongoDB supports horizontal scaling using **sharding** to distribute data across multiple servers.
Indexing – It supports various types of indexes to improve query performance.
Replication – Uses **replica sets** to ensure high availability and fault tolerance.
Aggregation Framework – Provides powerful data aggregation tools similar to SQL's `GROUP BY`.
High Performance – Efficient data storage and retrieval due to in-memory computing and indexing

B)What are Documents and Collections in MongoDB?

- **Documents**:
  The basic unit of data in MongoDB, stored in BSON format (similar to JSON).
  Example:

```
        {
  "_id": ObjectId("507f191e810c19729de860ea"),

  "name": "Alice",

  "age": 25,

  "email": "alice@example.com"

}
```

- **Collections**:
  A collection is a group of related documents (similar to a table in SQL).
  Unlike SQL tables, collections do not enforce a fixed schema.
  Example: A `users` collection can store multiple user documents.

C)When to use MongoDB?

Big Data Applications – When dealing with large volumes of unstructured or semi-structured data.

Real-Time Analytics – Due to its fast read/write capabilities and indexing.

Content Management Systems – Flexible schema makes it easy to store articles, blogs, and media content.

E-Commerce Applications – Can efficiently store products, user profiles, and transactions.

IoT and Mobile Apps – Handles high data ingestion and flexible data models.

Cloud-based Applications – Works well with distributed and scalable cloud environments.

D)What is Sharding in MongoDB?

Sharding is a method used in MongoDB to distribute data across multiple servers to ensure horizontal scalability.

It helps handle large datasets and high query loads.

How it works?

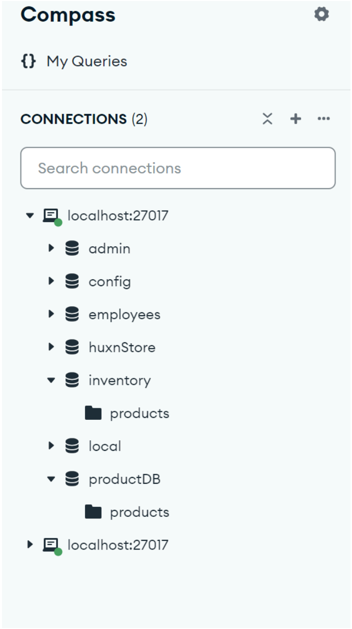
The dataset is partitioned into smaller pieces called **shards**.
A **shard key** determines how data is distributed across shards.
A **config server** manages metadata about the shards.
A **mongos router** directs queries to the appropriate shard.

Example of Sharding

If you have a large e-commerce database, you can shard it based on the user_id or product_category to distribute data efficiently across multiple servers.

Output:

Created a database 'inventory' with 'products' collection



Compass

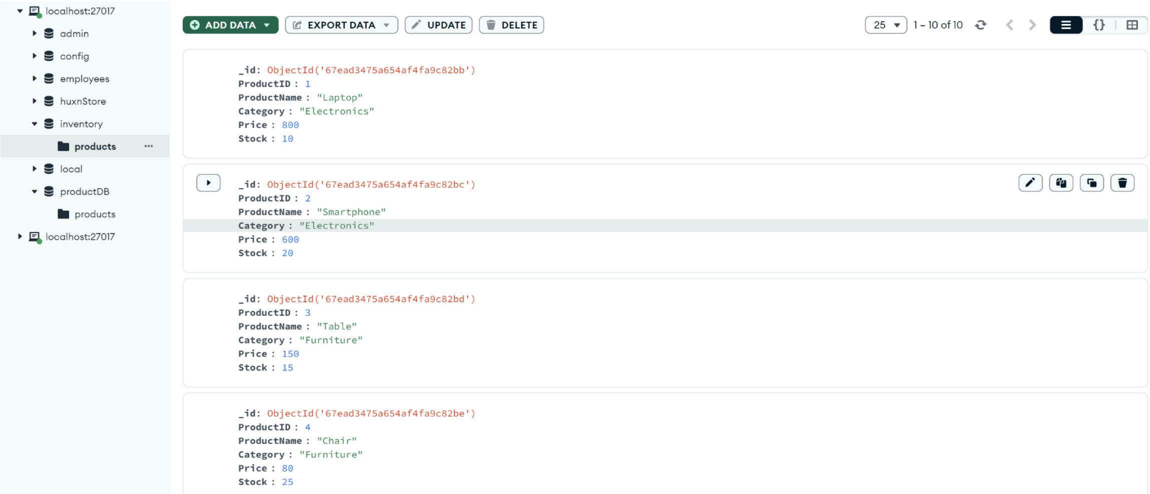
{} My Queries

CONNECTIONS (2)

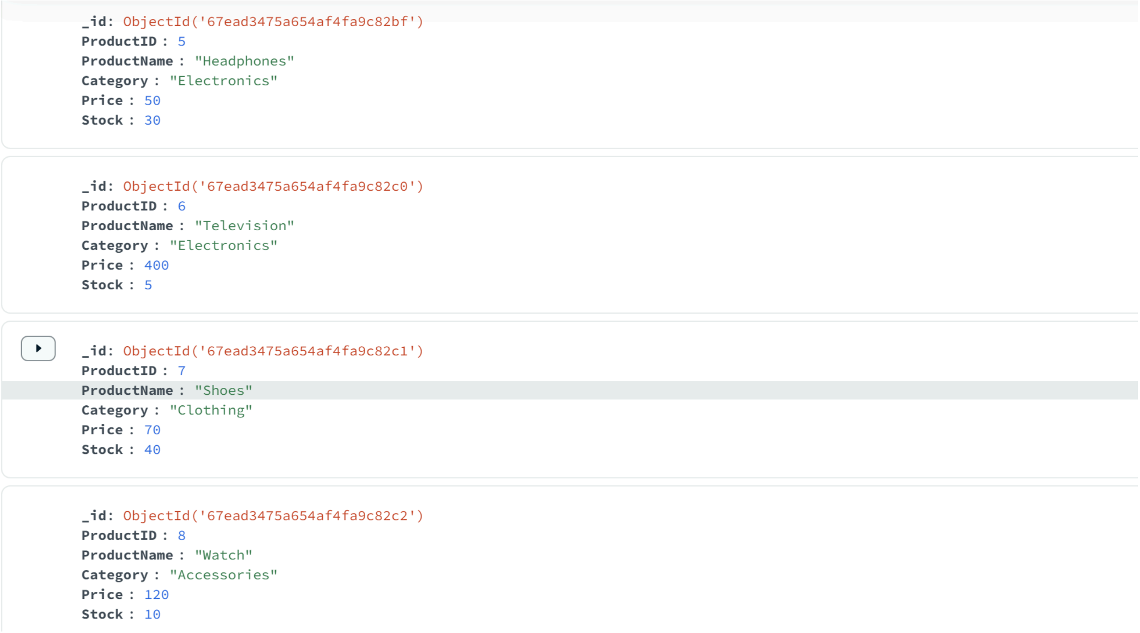
Search connections

- localhost:27017
  - admin
  - config
  - employees
  - huxnStore
  - inventory
    - products
  - local
  - productDB
    - products
- localhost:27017

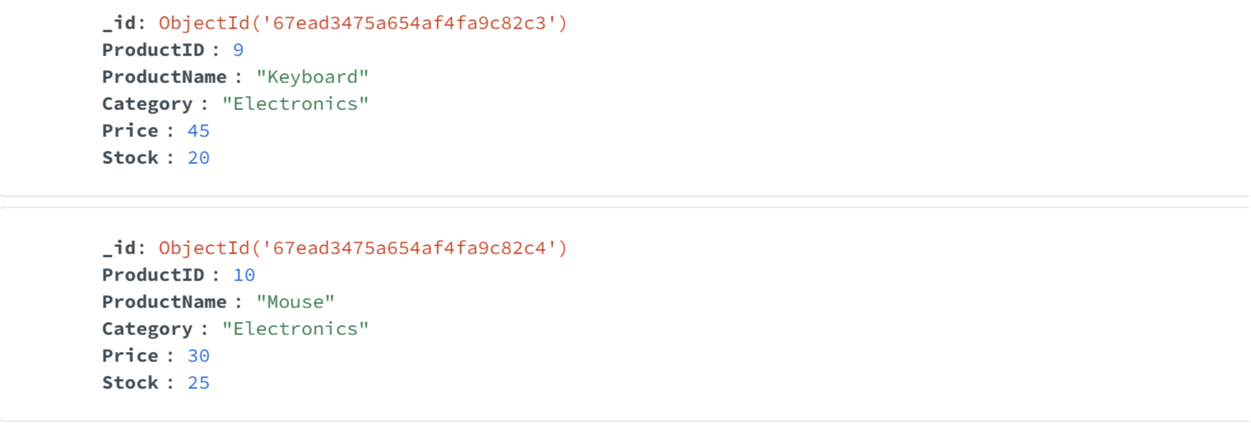
Inserted 10 documents into the "products" collection.



```
_id: ObjectId('67ead3475a654af4fa9c82bb')
ProductID : 1
ProductName : "Laptop"
Category : "Electronics"
Price : 800
Stock : 10

_id: ObjectId('67ead3475a654af4fa9c82bc')
ProductID : 2
ProductName : "Smartphone"
Category : "Electronics"
Price : 600
Stock : 20

_id: ObjectId('67ead3475a654af4fa9c82bd')
ProductID : 3
ProductName : "Table"
Category : "Furniture"
Price : 150
Stock : 15

_id: ObjectId('67ead3475a654af4fa9c82be')
ProductID : 4
ProductName : "Chair"
Category : "Furniture"
Price : 80
Stock : 25
```

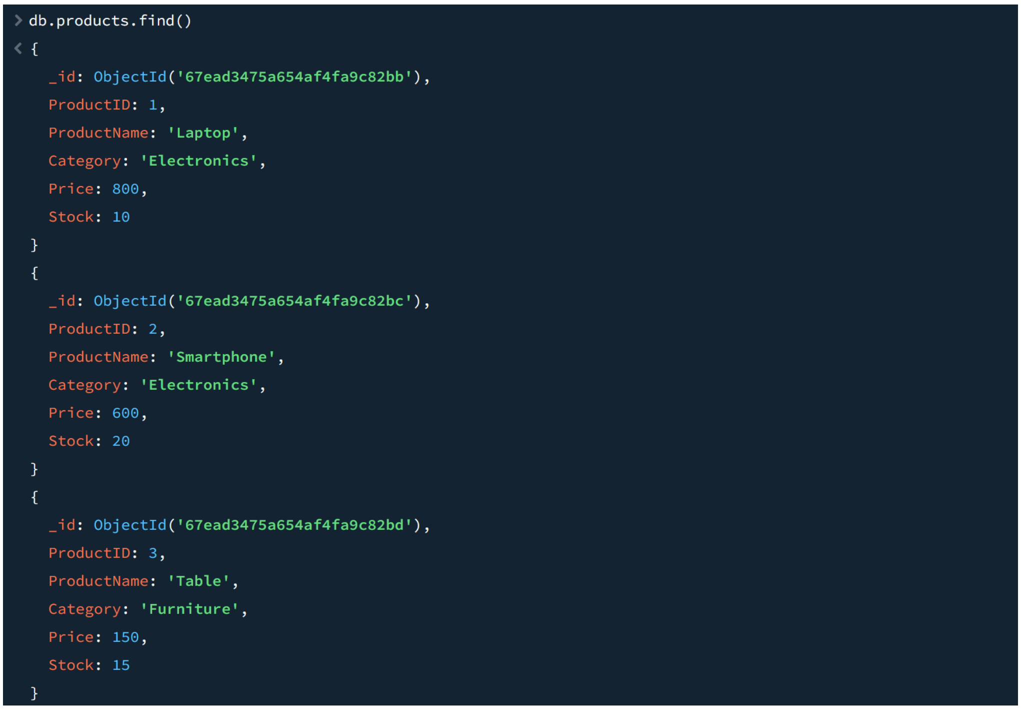


```
_id: ObjectId('67ead3475a654af4fa9c82bf')
ProductID : 5
ProductName : "Headphones"
Category : "Electronics"
Price : 50
Stock : 30

_id: ObjectId('67ead3475a654af4fa9c82c0')
ProductID : 6
ProductName : "Television"
Category : "Electronics"
Price : 400
Stock : 5

_id: ObjectId('67ead3475a654af4fa9c82c1')
ProductID : 7
ProductName : "Shoes"
Category : "Clothing"
Price : 70
Stock : 40

_id: ObjectId('67ead3475a654af4fa9c82c2')
ProductID : 8
ProductName : "Watch"
Category : "Accessories"
Price : 120
Stock : 10
```

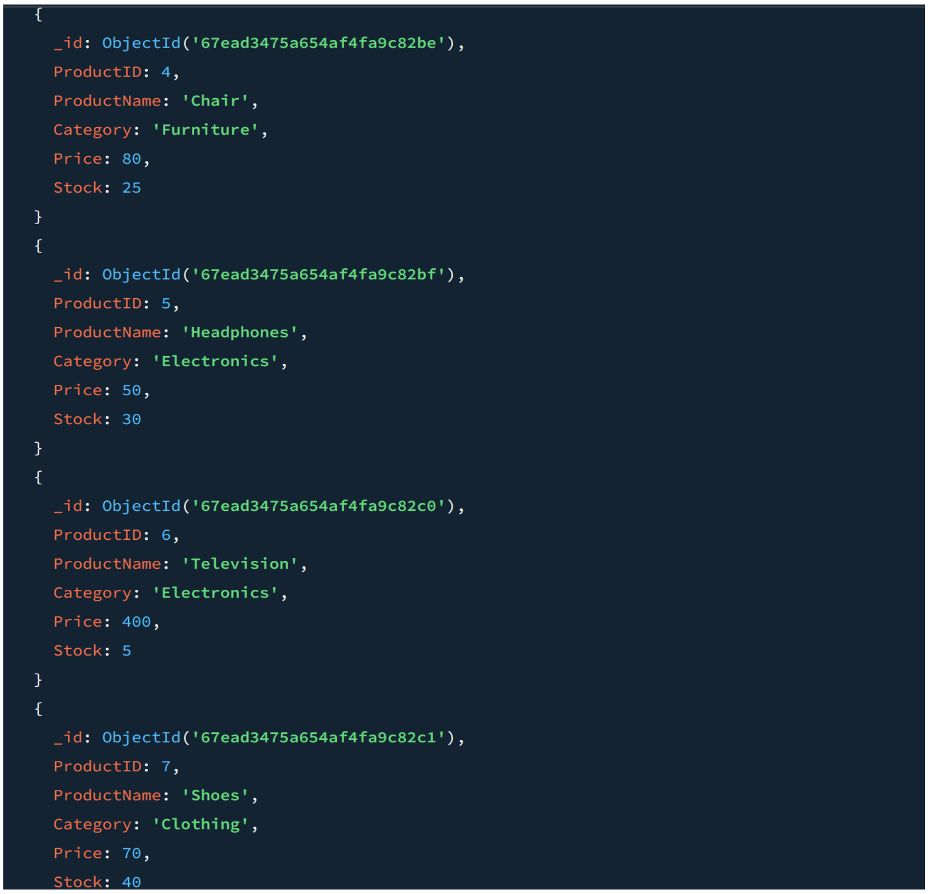


```
_id: ObjectId('67ead3475a654af4fa9c82c3')
ProductID : 9
ProductName : "Keyboard"
Category : "Electronics"
Price : 45
Stock : 20

_id: ObjectId('67ead3475a654af4fa9c82c4')
ProductID : 10
ProductName : "Mouse"
Category : "Electronics"
Price : 30
Stock : 25
```

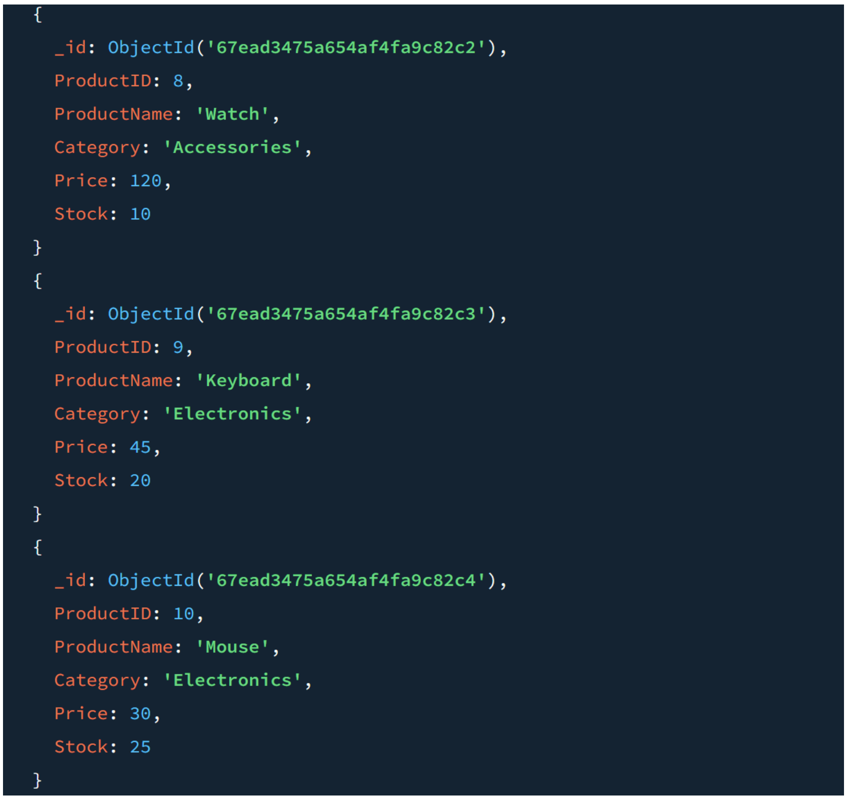
1. Display all documents in the "products" collection



```
> db.products.find()
< {
    _id: ObjectId('67ead3475a654af4fa9c82bb'),
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 800,
    Stock: 10
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bc'),
    ProductID: 2,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 600,
    Stock: 20
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bd'),
    ProductID: 3,
    ProductName: 'Table',
    Category: 'Furniture',
    Price: 150,
    Stock: 15
  }
```

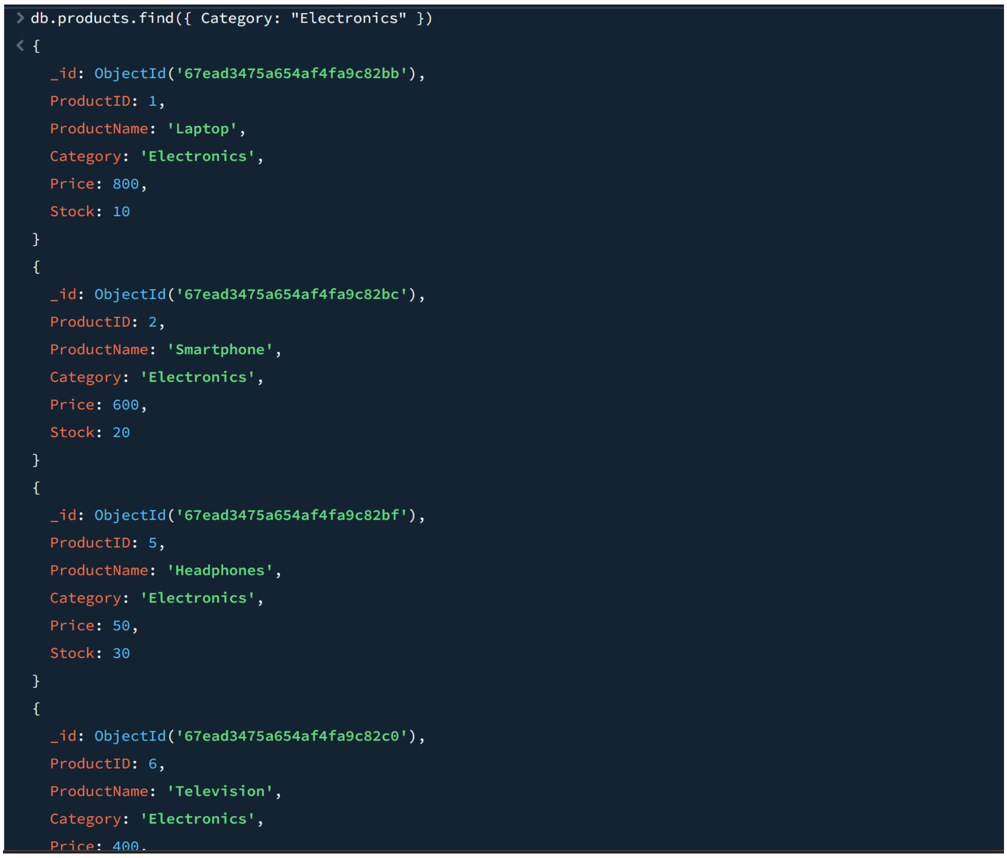


```
  {
    _id: ObjectId('67ead3475a654af4fa9c82be'),
    ProductID: 4,
    ProductName: 'Chair',
    Category: 'Furniture',
    Price: 80,
    Stock: 25
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bf'),
    ProductID: 5,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 50,
    Stock: 30
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82c0'),
    ProductID: 6,
    ProductName: 'Television',
    Category: 'Electronics',
    Price: 400,
    Stock: 5
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82c1'),
    ProductID: 7,
    ProductName: 'Shoes',
    Category: 'Clothing',
    Price: 70,
    Stock: 40
```

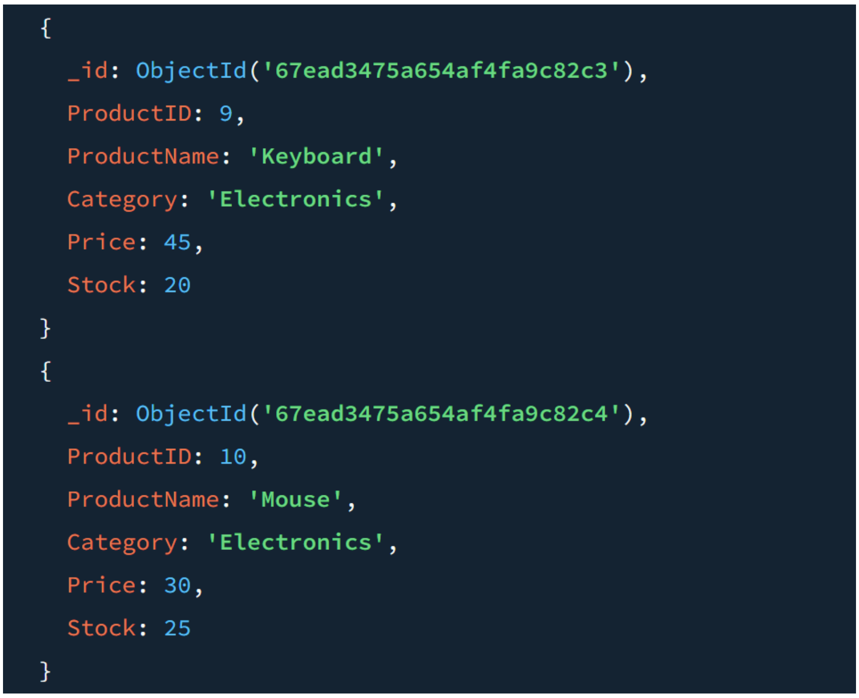


```
  {
    _id: ObjectId('67ead3475a654af4fa9c82c2'),
    ProductID: 8,
    ProductName: 'Watch',
    Category: 'Accessories',
    Price: 120,
    Stock: 10
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82c3'),
    ProductID: 9,
    ProductName: 'Keyboard',
    Category: 'Electronics',
    Price: 45,
    Stock: 20
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82c4'),
    ProductID: 10,
    ProductName: 'Mouse',
    Category: 'Electronics',
    Price: 30,
    Stock: 25
  }
```

2.Display all products in the "Electronics" category

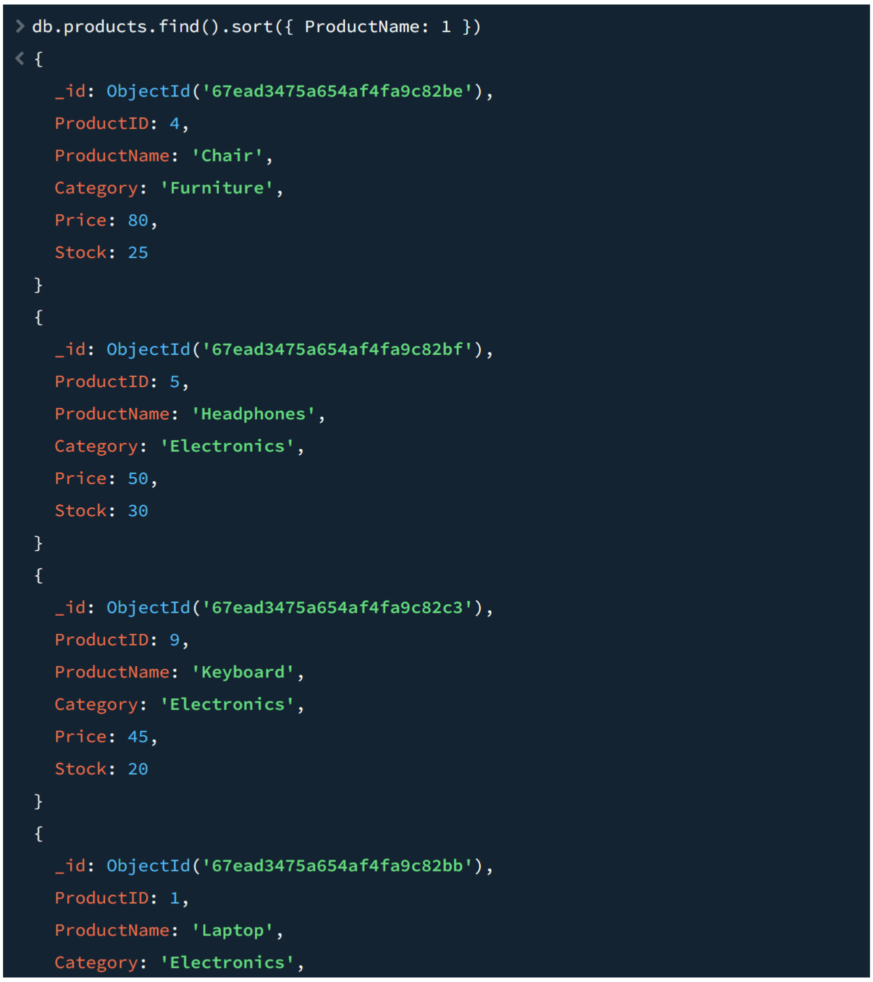


```
> db.products.find({ Category: "Electronics" })
< {
    _id: ObjectId('67ead3475a654af4fa9c82bb'),
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 800,
    Stock: 10
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bc'),
    ProductID: 2,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 600,
    Stock: 20
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bf'),
    ProductID: 5,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 50,
    Stock: 30
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82c0'),
    ProductID: 6,
    ProductName: 'Television',
    Category: 'Electronics',
    Price: 400,
```

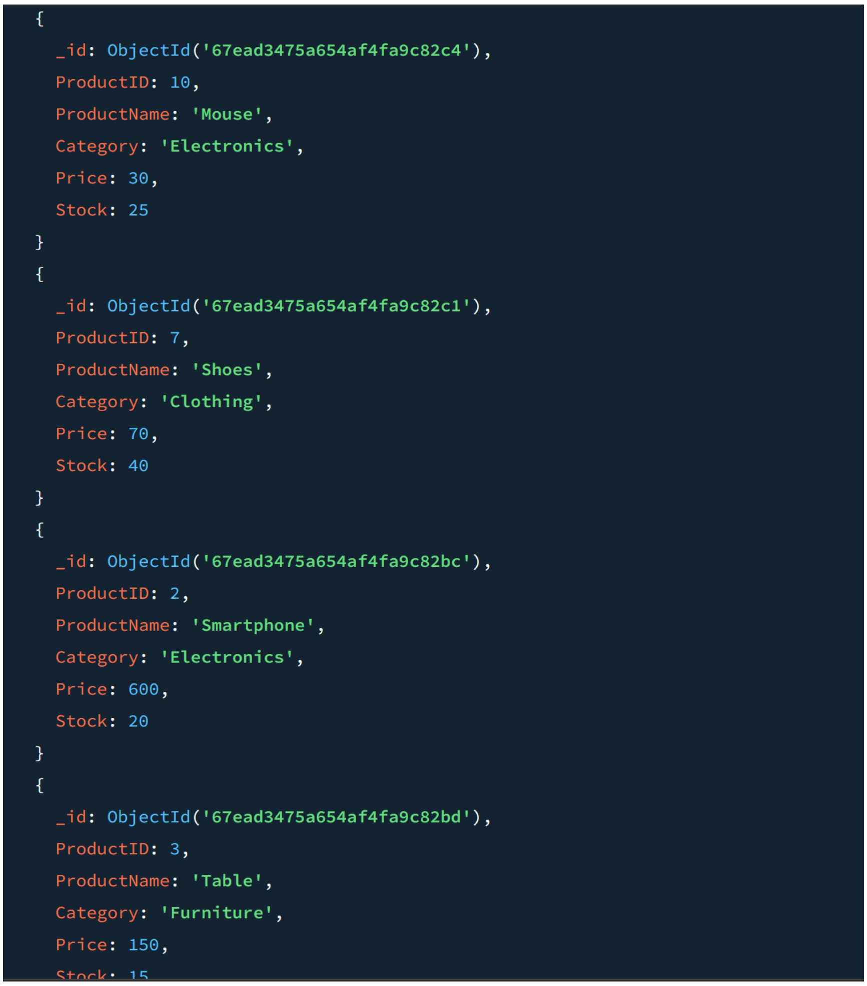


```
{
  _id: ObjectId('67ead3475a654af4fa9c82c3'),
  ProductID: 9,
  ProductName: 'Keyboard',
  Category: 'Electronics',
  Price: 45,
  Stock: 20
}
{
  _id: ObjectId('67ead3475a654af4fa9c82c4'),
  ProductID: 10,
  ProductName: 'Mouse',
  Category: 'Electronics',
  Price: 30,
  Stock: 25
}
```

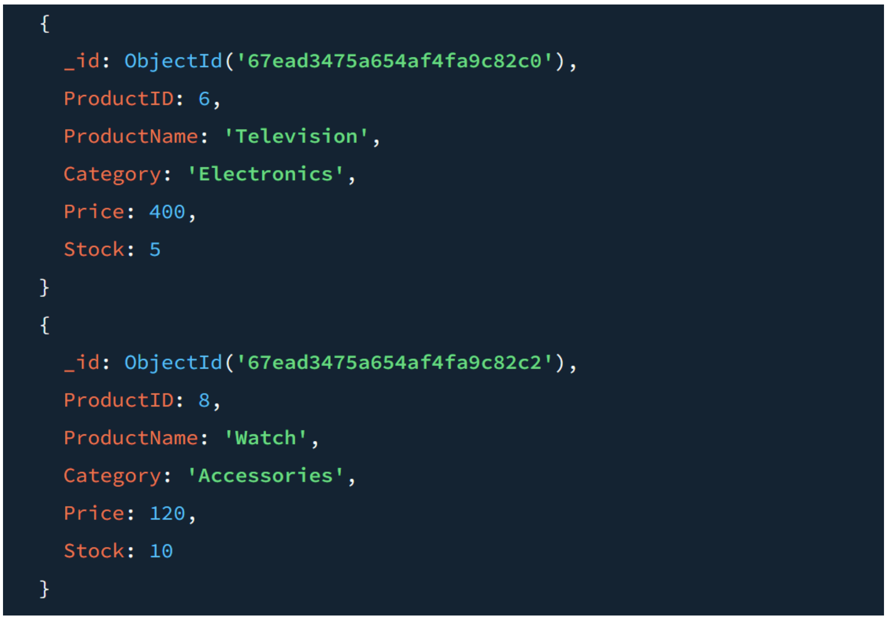
3.Display all products in ascending order of their names



```
> db.products.find().sort({ ProductName: 1 })
< {
    _id: ObjectId('67ead3475a654af4fa9c82be'),
    ProductID: 4,
    ProductName: 'Chair',
    Category: 'Furniture',
    Price: 80,
    Stock: 25
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bf'),
    ProductID: 5,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 50,
    Stock: 30
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82c3'),
    ProductID: 9,
    ProductName: 'Keyboard',
    Category: 'Electronics',
    Price: 45,
    Stock: 20
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bb'),
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
```

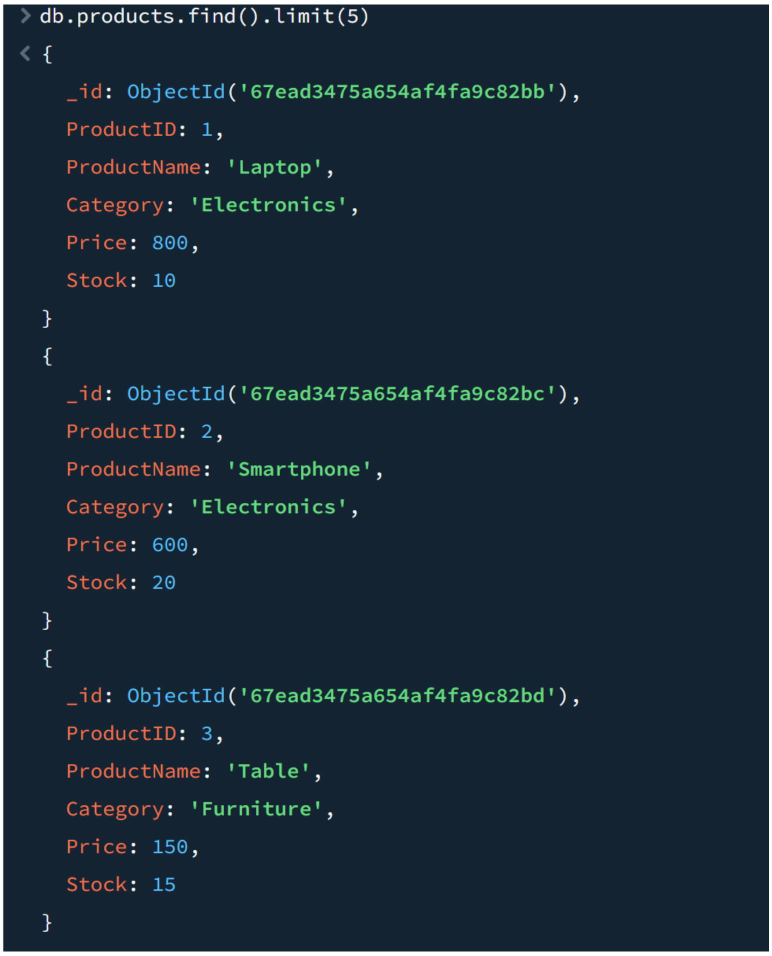


```
{
  _id: ObjectId('67ead3475a654af4fa9c82c4'),
  ProductID: 10,
  ProductName: 'Mouse',
  Category: 'Electronics',
  Price: 30,
  Stock: 25
}
{
  _id: ObjectId('67ead3475a654af4fa9c82c1'),
  ProductID: 7,
  ProductName: 'Shoes',
  Category: 'Clothing',
  Price: 70,
  Stock: 40
}
{
  _id: ObjectId('67ead3475a654af4fa9c82bc'),
  ProductID: 2,
  ProductName: 'Smartphone',
  Category: 'Electronics',
  Price: 600,
  Stock: 20
}
{
  _id: ObjectId('67ead3475a654af4fa9c82bd'),
  ProductID: 3,
  ProductName: 'Table',
  Category: 'Furniture',
  Price: 150,
  Stock: 15
```

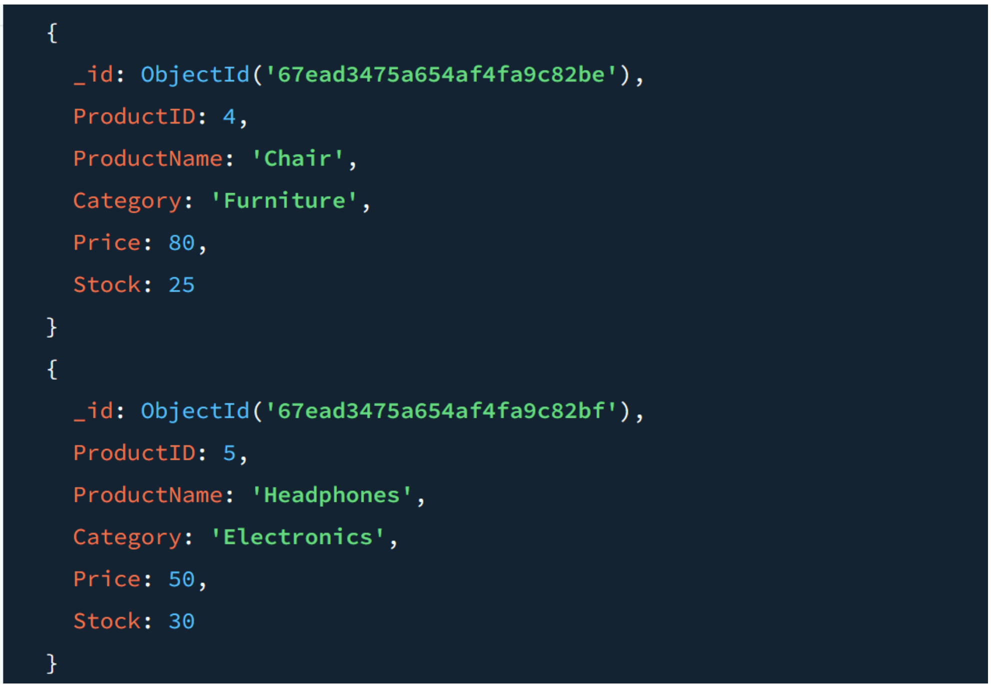


```
{
  _id: ObjectId('67ead3475a654af4fa9c82c0'),
  ProductID: 6,
  ProductName: 'Television',
  Category: 'Electronics',
  Price: 400,
  Stock: 5
}
{
  _id: ObjectId('67ead3475a654af4fa9c82c2'),
  ProductID: 8,
  ProductName: 'Watch',
  Category: 'Accessories',
  Price: 120,
  Stock: 10
}
```

4. Display the details of the first 5 products

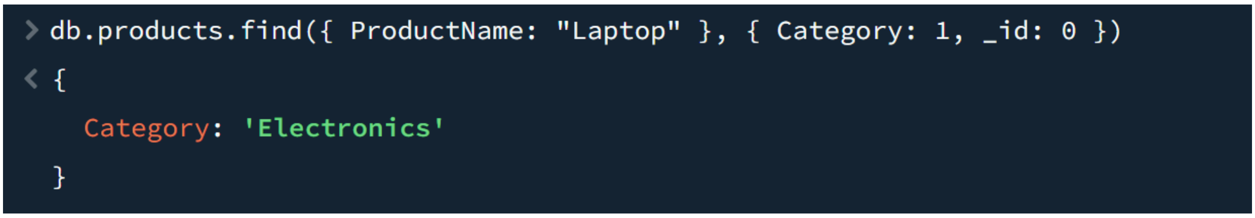


```
> db.products.find().limit(5)
< {
    _id: ObjectId('67ead3475a654af4fa9c82bb'),
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 800,
    Stock: 10
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bc'),
    ProductID: 2,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 600,
    Stock: 20
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bd'),
    ProductID: 3,
    ProductName: 'Table',
    Category: 'Furniture',
    Price: 150,
    Stock: 15
  }
```



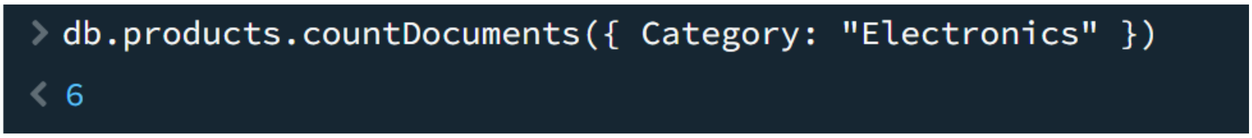
```
  {
    _id: ObjectId('67ead3475a654af4fa9c82be'),
    ProductID: 4,
    ProductName: 'Chair',
    Category: 'Furniture',
    Price: 80,
    Stock: 25
  }
  {
    _id: ObjectId('67ead3475a654af4fa9c82bf'),
    ProductID: 5,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 50,
    Stock: 30
  }
```

5.Display the categories of products with a specific name (e.g., "Laptop")



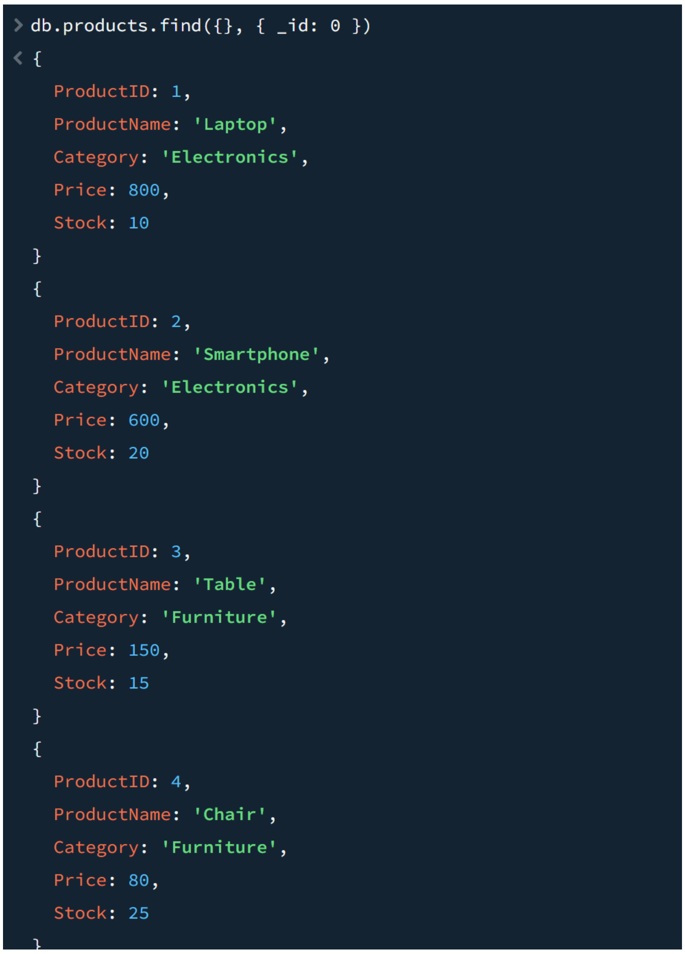
```
> db.products.find({ ProductName: "Laptop" }, { Category: 1, _id: 0 })
< {
    Category: 'Electronics'
  }
```

6.Display the number of products in the "Electronics" category

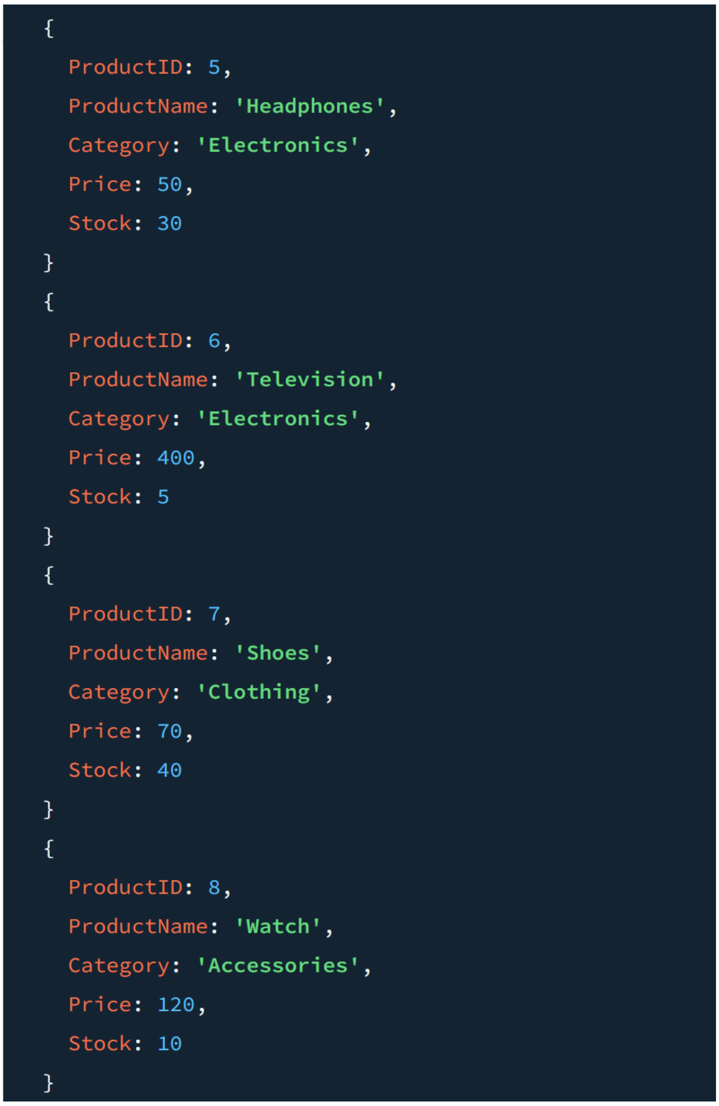


```
> db.products.countDocuments({ Category: "Electronics" })
< 6
```

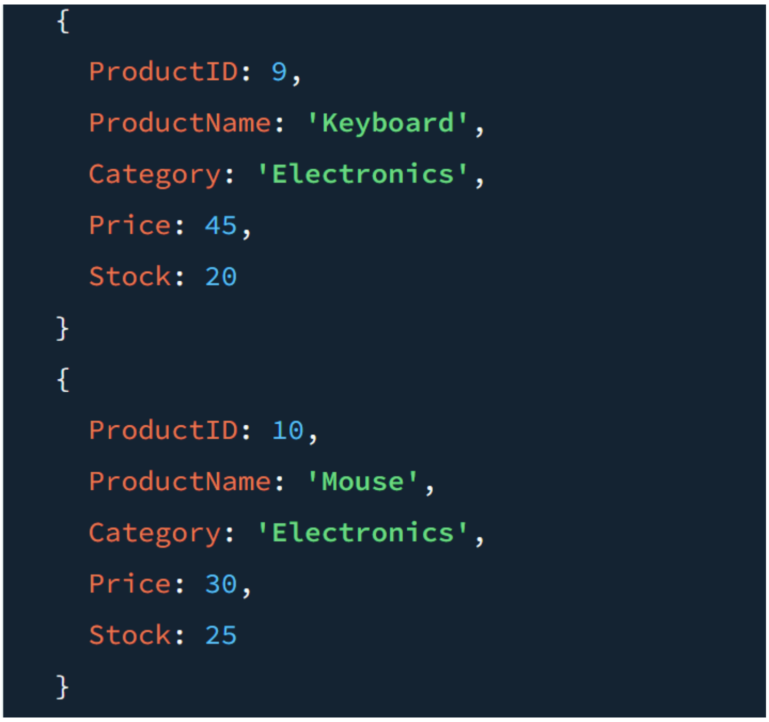
7.Display all products without showing the "_id" field



```
> db.products.find({}, { _id: 0 })
< {
    ProductID: 1,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 800,
    Stock: 10
  }
  {
    ProductID: 2,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 600,
    Stock: 20
  }
  {
    ProductID: 3,
    ProductName: 'Table',
    Category: 'Furniture',
    Price: 150,
    Stock: 15
  }
  {
    ProductID: 4,
    ProductName: 'Chair',
    Category: 'Furniture',
    Price: 80,
    Stock: 25
  }
```

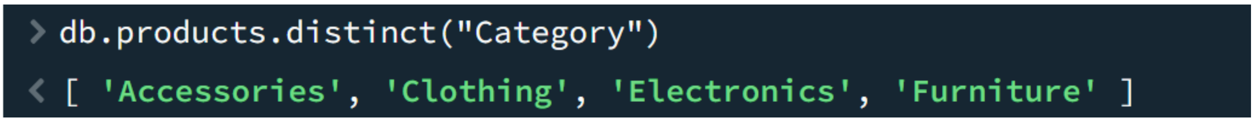


```
  {
    ProductID: 5,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 50,
    Stock: 30
  }
  {
    ProductID: 6,
    ProductName: 'Television',
    Category: 'Electronics',
    Price: 400,
    Stock: 5
  }
  {
    ProductID: 7,
    ProductName: 'Shoes',
    Category: 'Clothing',
    Price: 70,
    Stock: 40
  }
  {
    ProductID: 8,
    ProductName: 'Watch',
    Category: 'Accessories',
    Price: 120,
    Stock: 10
  }
```



```
  {
    ProductID: 9,
    ProductName: 'Keyboard',
    Category: 'Electronics',
    Price: 45,
    Stock: 20
  }
  {
    ProductID: 10,
    ProductName: 'Mouse',
    Category: 'Electronics',
    Price: 30,
    Stock: 25
  }
```

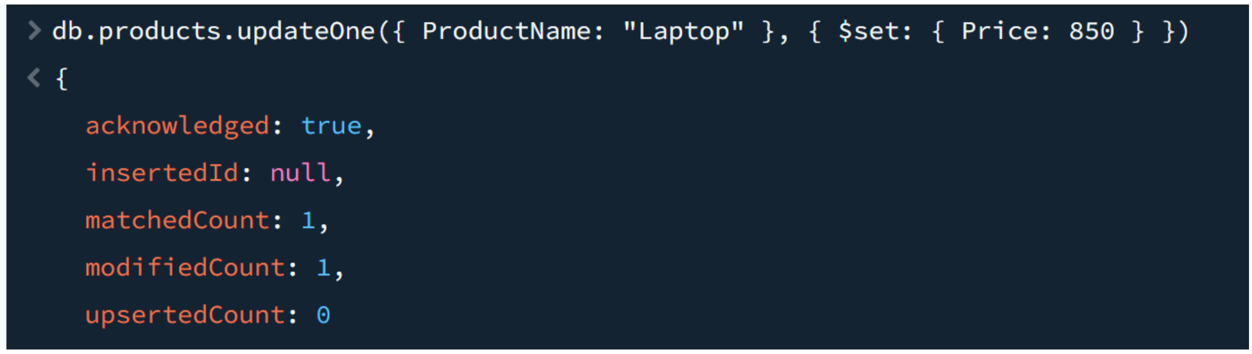
8. Display all distinct categories of products



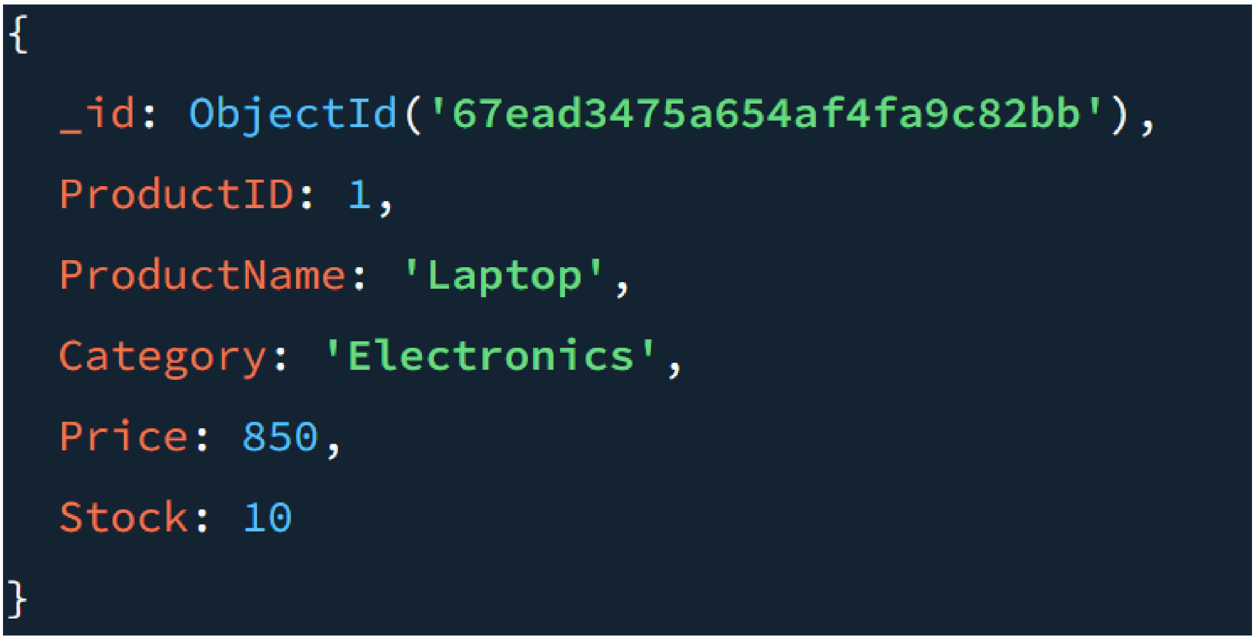
```
> db.products.distinct("Category")
< [ 'Accessories', 'Clothing', 'Electronics', 'Furniture' ]
```

9.Display products in the "Electronics" category with prices greater than 50 but less than 100

10.Change the price of a product (e.g., change the price of "Laptop" to 850)

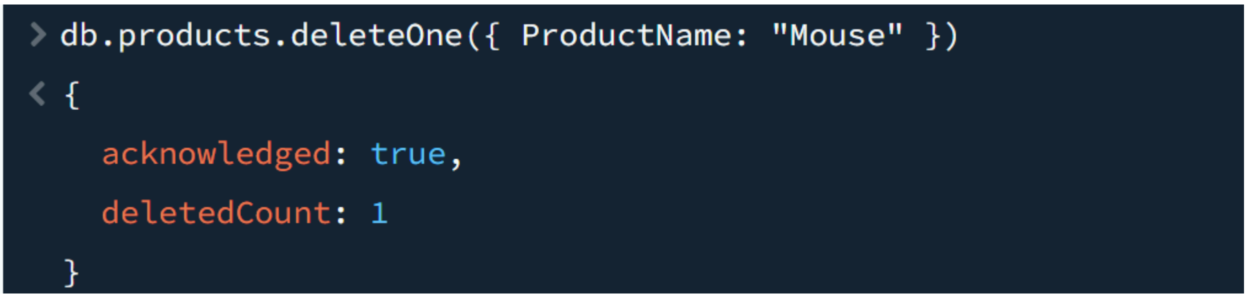


```
> db.products.updateOne({ ProductName: "Laptop" }, { $set: { Price: 850 } })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
```



```
{
  _id: ObjectId('67ead3475a654af4fa9c82bb'),
  ProductID: 1,
  ProductName: 'Laptop',
  Category: 'Electronics',
  Price: 850,
  Stock: 10
}
```

11. Delete a particular product entry (e.g., delete "Mouse")



```
> db.products.deleteOne({ ProductName: "Mouse" })
< {
    acknowledged: true,
    deletedCount: 1
  }
```