Experiment – 6: MongoDB

Name of Student	**Kartik bhat**
Class Roll No	**D15A / 03**
D.O.P.	
D.O.S.	
Sign and Grade	

Aim: To study CRUD operations in MongoDB.

Problem Statement:

A) 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?

• Document-Oriented: Stores data as flexible, JSON-like documents (BSON). • Flexible Schema: No fixed structure, supports dynamic data.
• Horizontal Scalability: Uses sharding to manage large datasets.
• Replication: Ensures high availability with replica sets.
• Indexing: Supports various indexes for faster query execution.
• Aggregation Framework: Provides powerful data processing using pipelines. • Ad-hoc Queries: Enables complex queries with ease

b. What are Documents and Collections in MongoDB?

Documents: JSON-like records storing data in key-value pairs. Example:

```
{
    "_id": "101",
    "name": "Alice",
    "age": 28,
    "email": alice@example.com
}
```

Collections: A group of documents, equivalent to tables in relational databases. They don't enforce strict schemas, allowing flexibility.

c. When to use MongoDB?

• Big Data Applications: Efficient for large, unstructured data.
• E-commerce Platforms: Ideal for product catalogs with dynamic attributes. • Content Management Systems (CMS): Supports frequent changes in data models.
• Real-Time Analytics: Processes and analyzes data rapidly.
• IoT and Mobile Apps: Manages sensor data and app data effectively. • Social Networks: Scales well for user-generated content.

d. What is Sharding in MongoDB?

Sharding: Distributes data across multiple servers to handle large datasets.
Shard Key: A field in documents used to split data across shards.

Components:

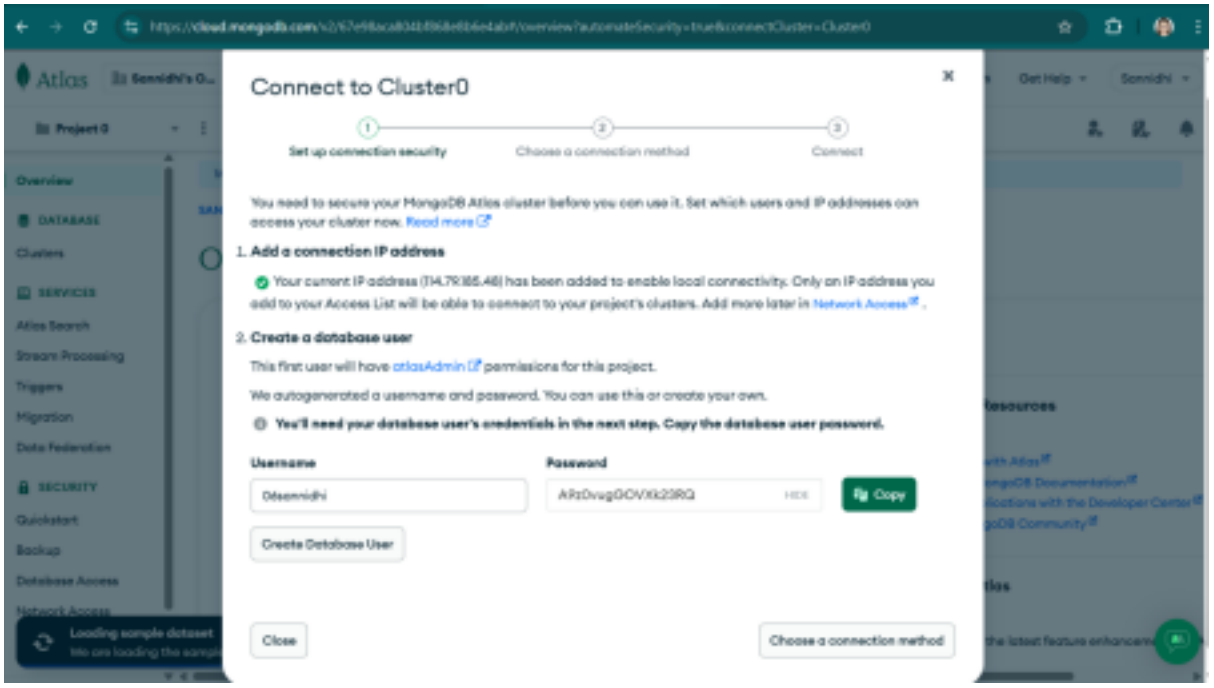
- Shards: Store actual data.
- Config Servers: Maintain metadata and sharding configuration.
- Mongos: Routes queries to the appropriate shards.

Benefits:

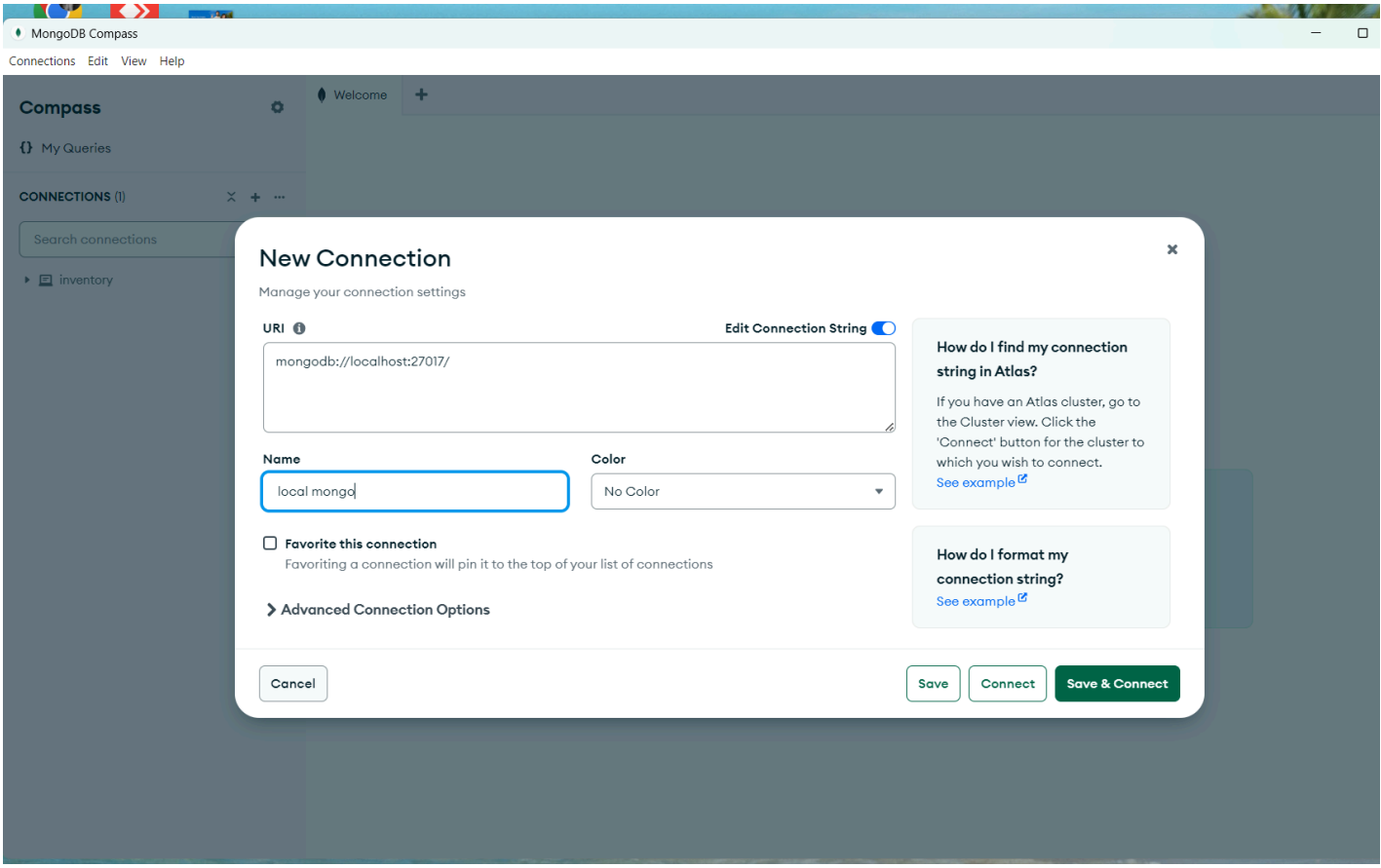
- Supports large-scale data management.
- Improves read and write performance.
- Ensures fault tolerance and high availability.

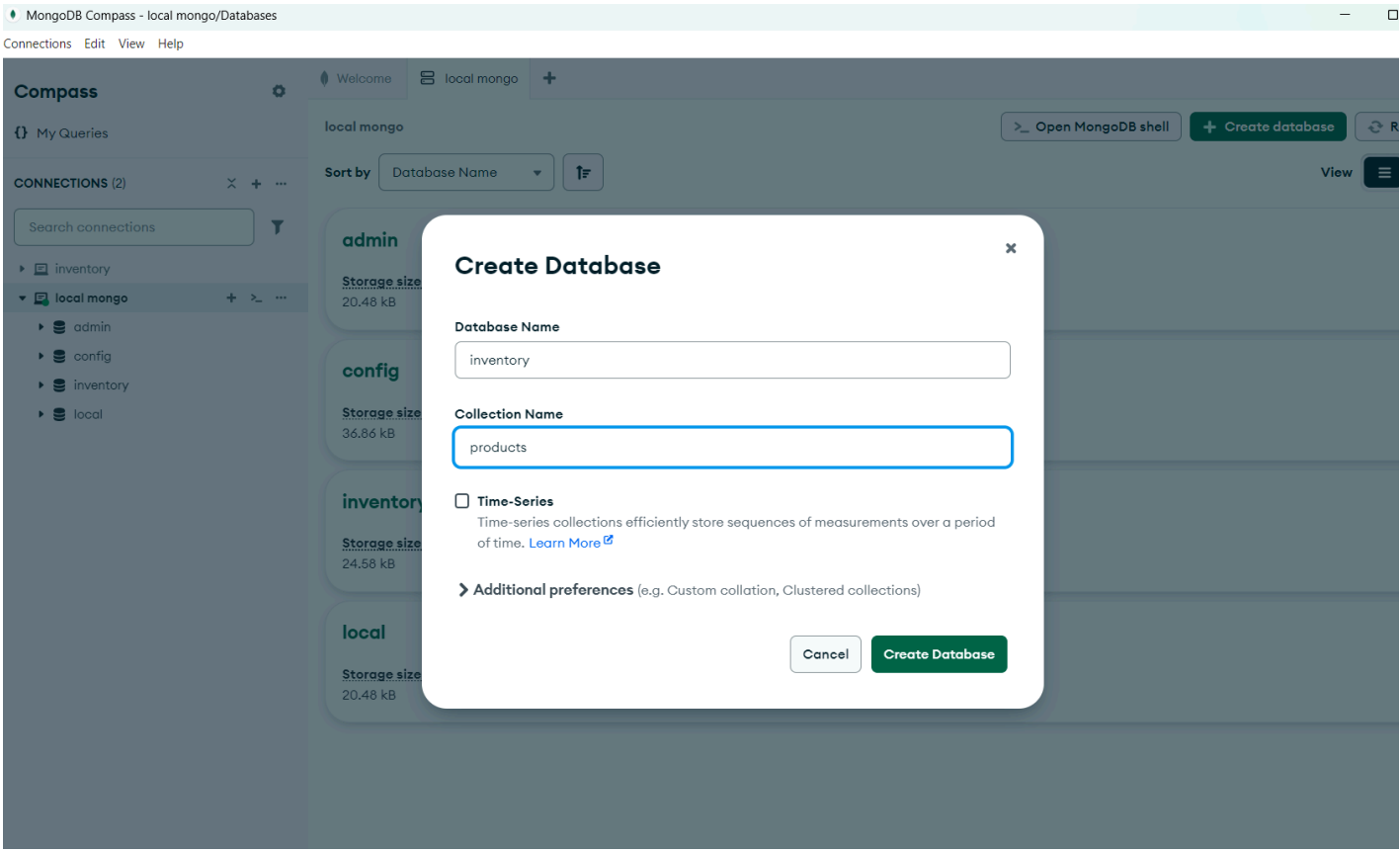
Output:

Connecting MongoDB Compass to Atlas Cluster

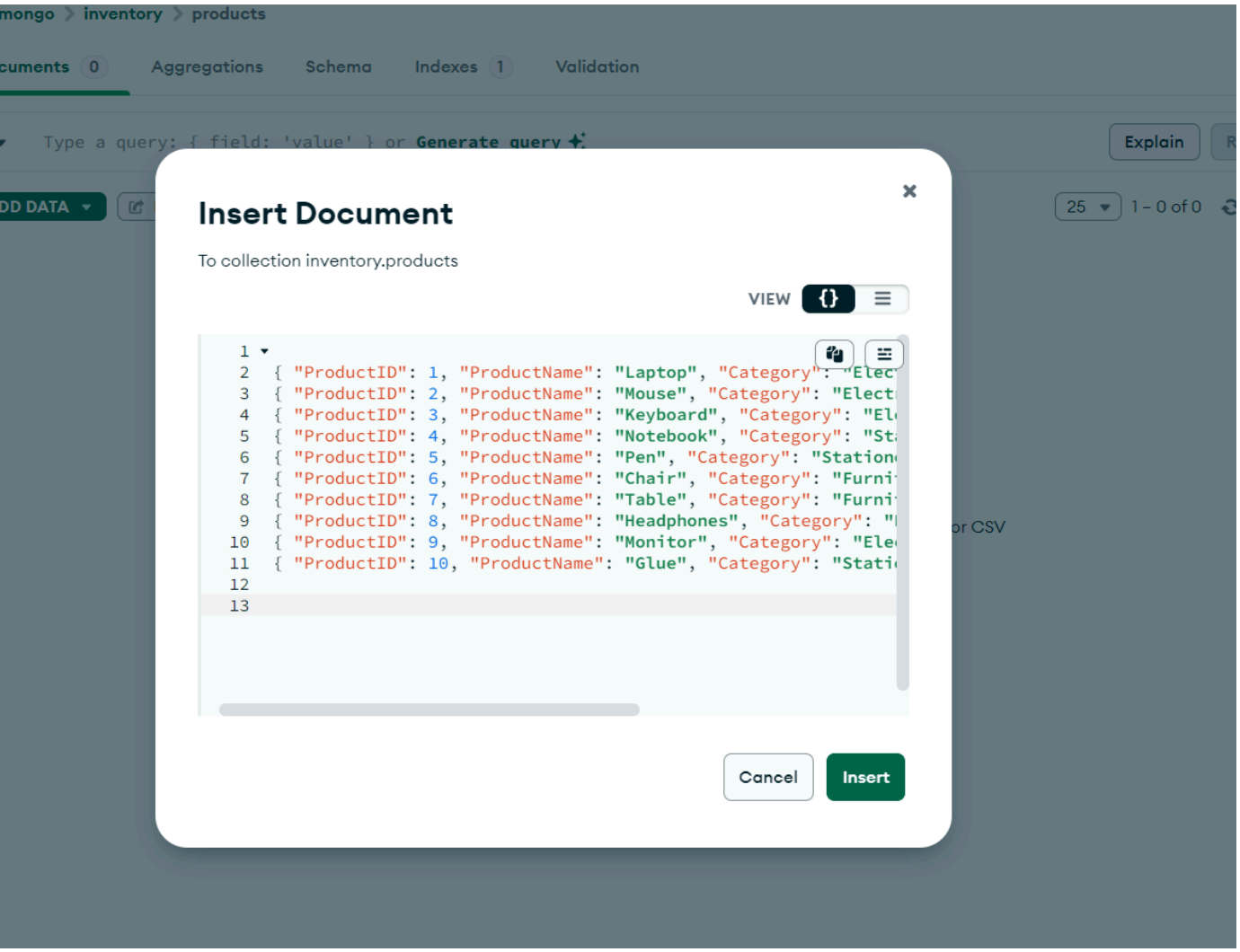


Create a Database and Collection

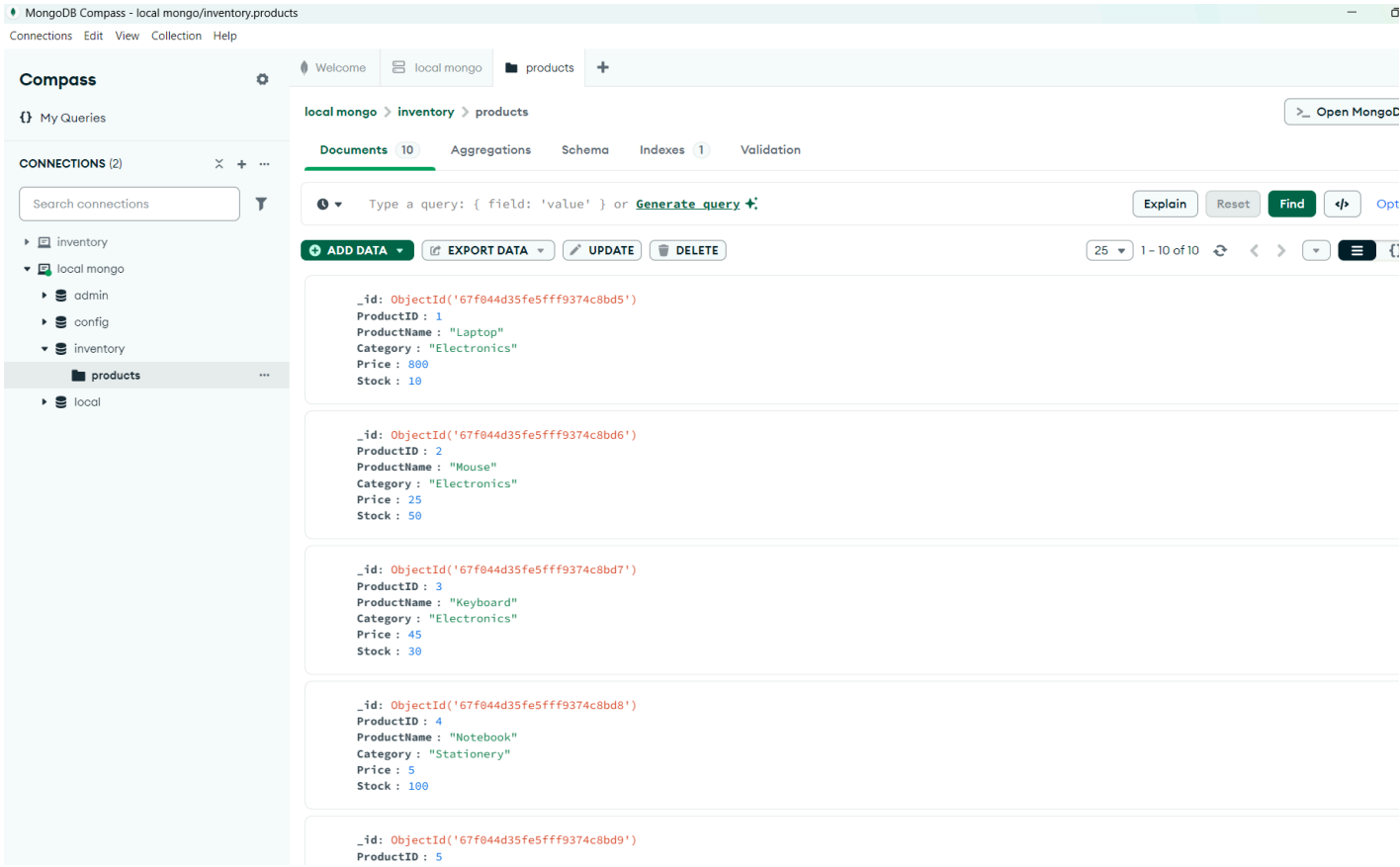




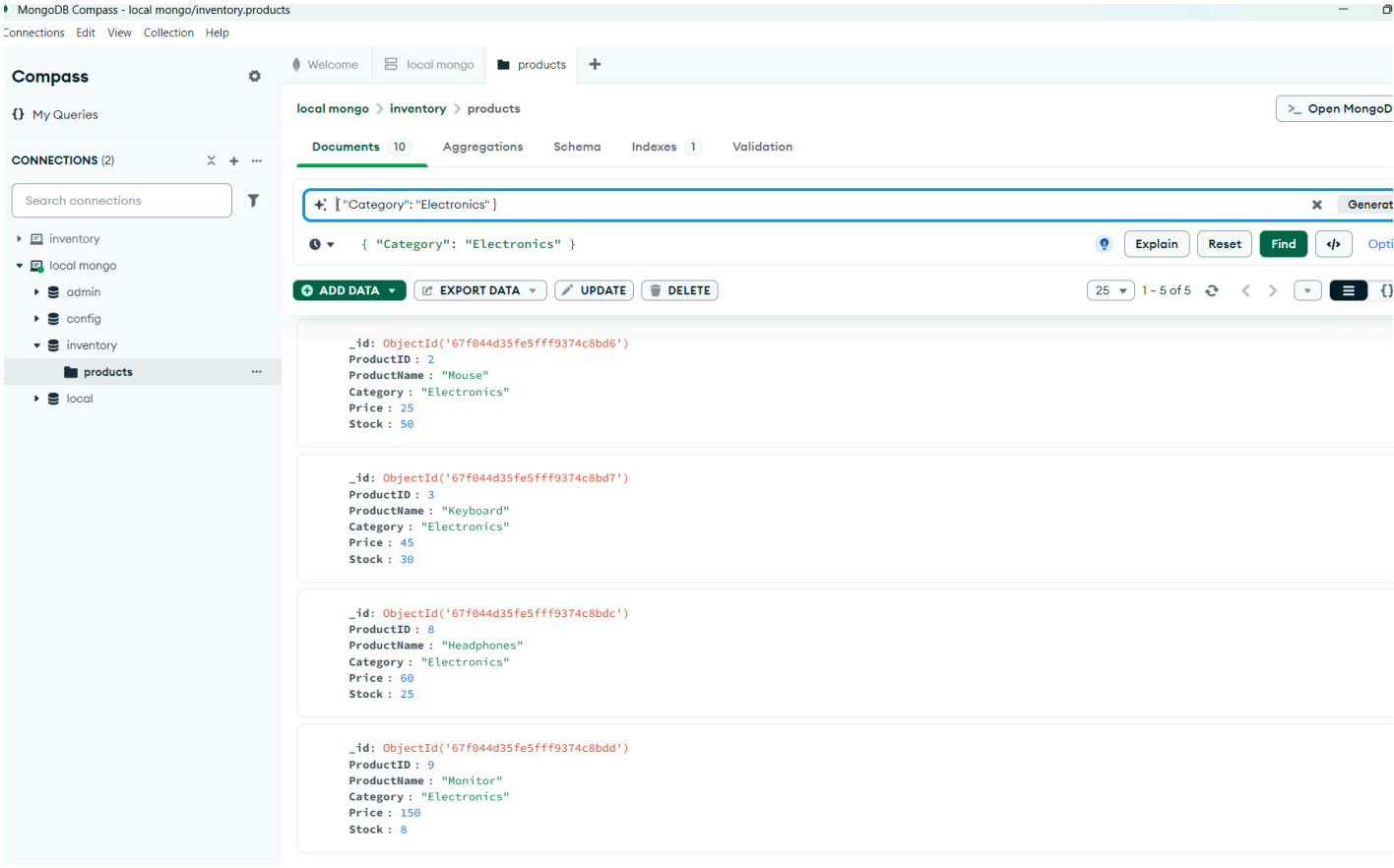
Insert 10 Documents into "products" Collection



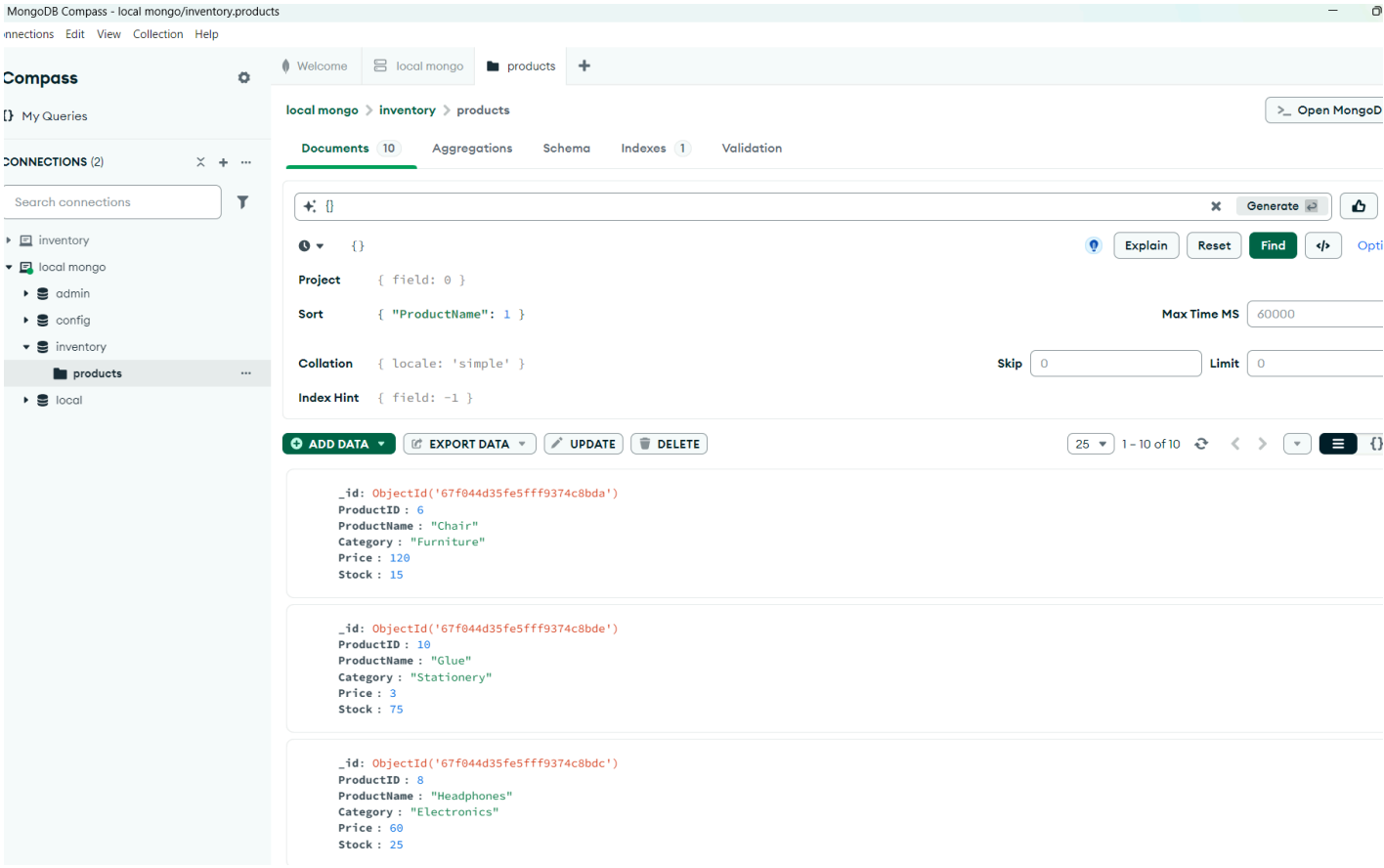
Display All Documents in "products" Collection



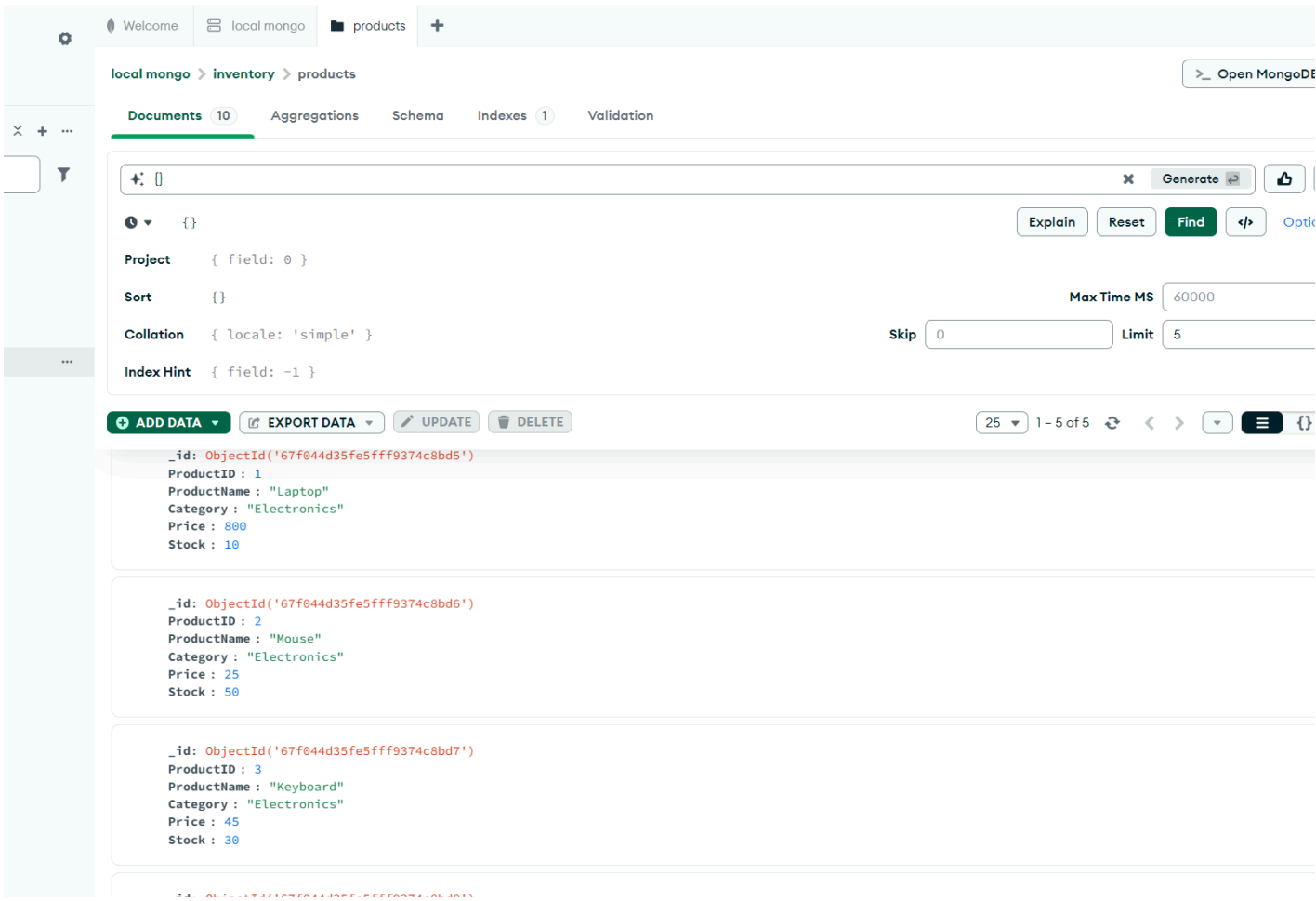
Display All Products in the "Electronics" Category


MongoDB Compass - local mongo/inventory.products
Connections Edit View Collection Help
Compass
My Queries
CONNECTIONS (2)
Search connections
inventory
local mongo
admin
config
inventory
products
local
Welcome
local mongo
products
local mongo > inventory > products
Open MongoD
Documents 10
Aggregations
Schema
Indexes 1
Validation
{"Category": "Electronics"}
Generat
{ "Category": "Electronics" }
Explain
Reset
Find
Opti
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 5 of 5
_id: ObjectId('67f044d35fe5fff9374c8bd6')
ProductID : 2
ProductName : "Mouse"
Category : "Electronics"
Price : 25
Stock : 50
_id: ObjectId('67f044d35fe5fff9374c8bd7')
ProductID : 3
ProductName : "Keyboard"
Category : "Electronics"
Price : 45
Stock : 30
_id: ObjectId('67f044d35fe5fff9374c8bdc')
ProductID : 8
ProductName : "Headphones"
Category : "Electronics"
Price : 60
Stock : 25
_id: ObjectId('67f044d35fe5fff9374c8bdd')
ProductID : 9
ProductName : "Monitor"
Category : "Electronics"
Price : 150
Stock : 8

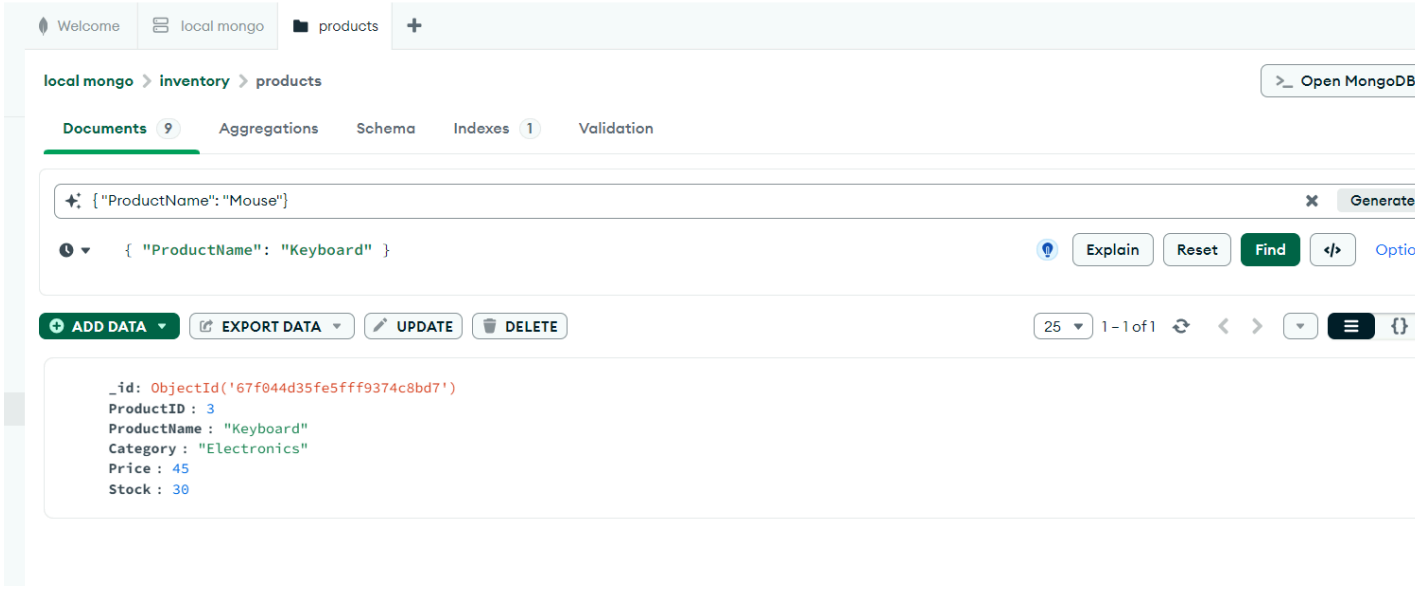
Display All Products in Ascending Order of Their Name



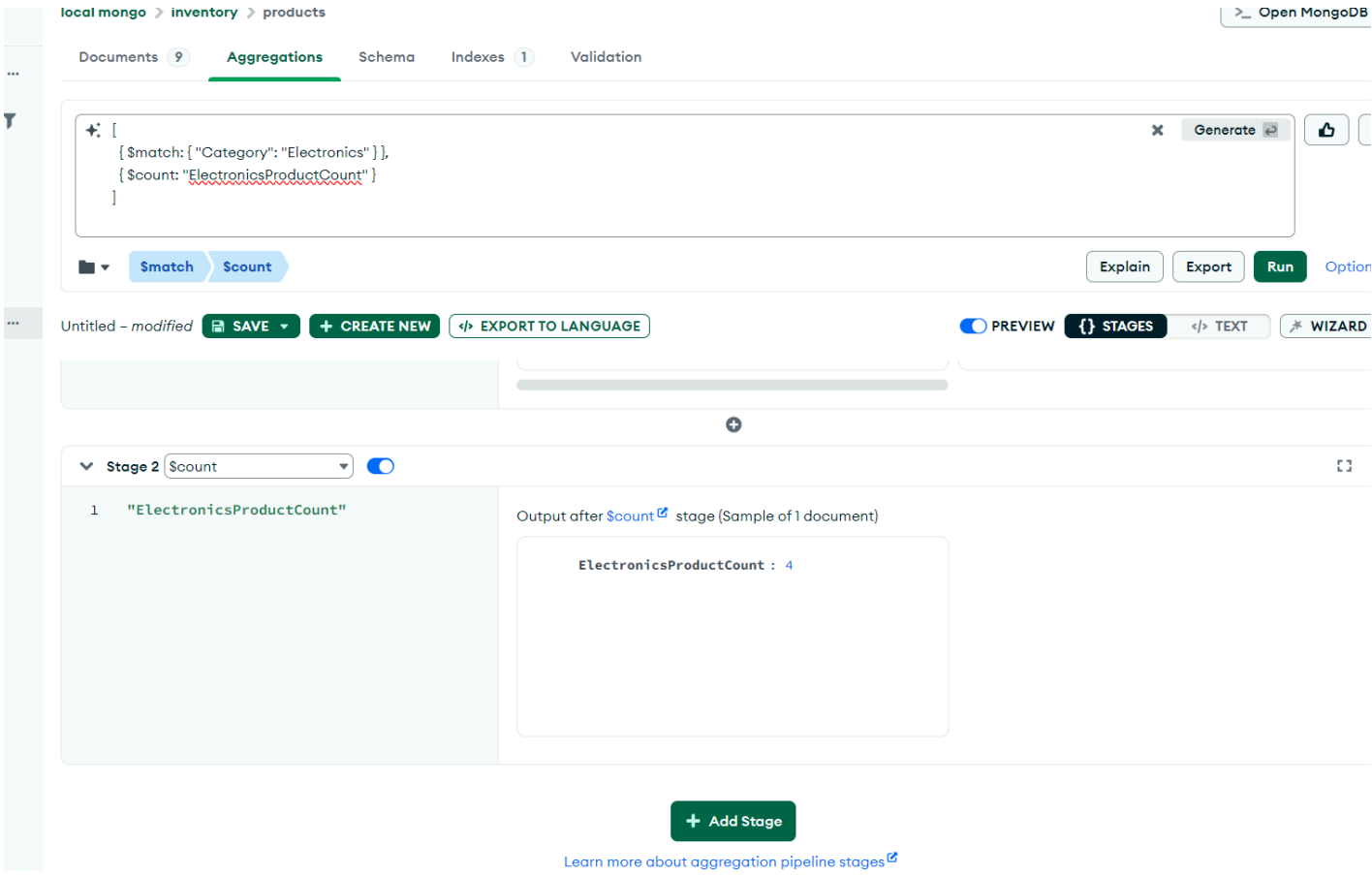
Display the First 5 Products



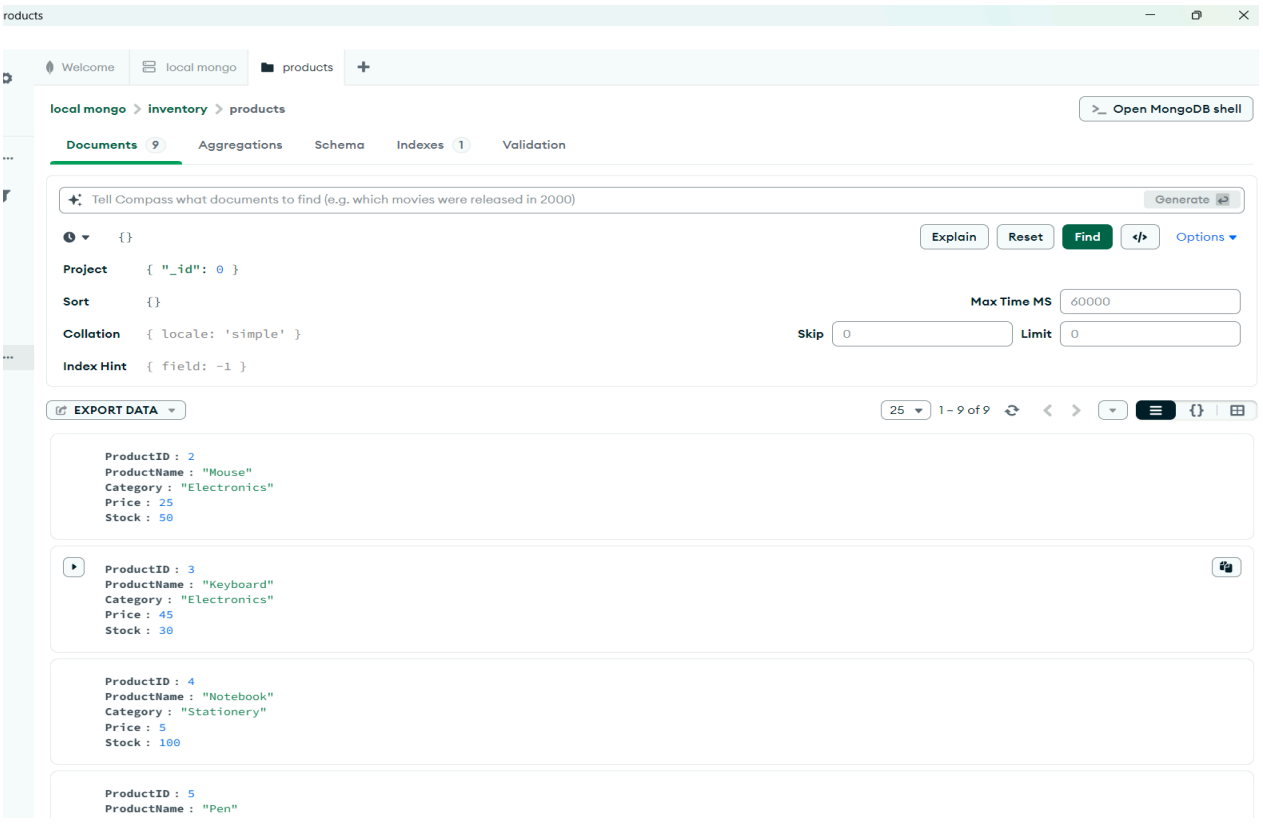
Display the Categories of Products with a Specific Name (e.g., "Keyboard")



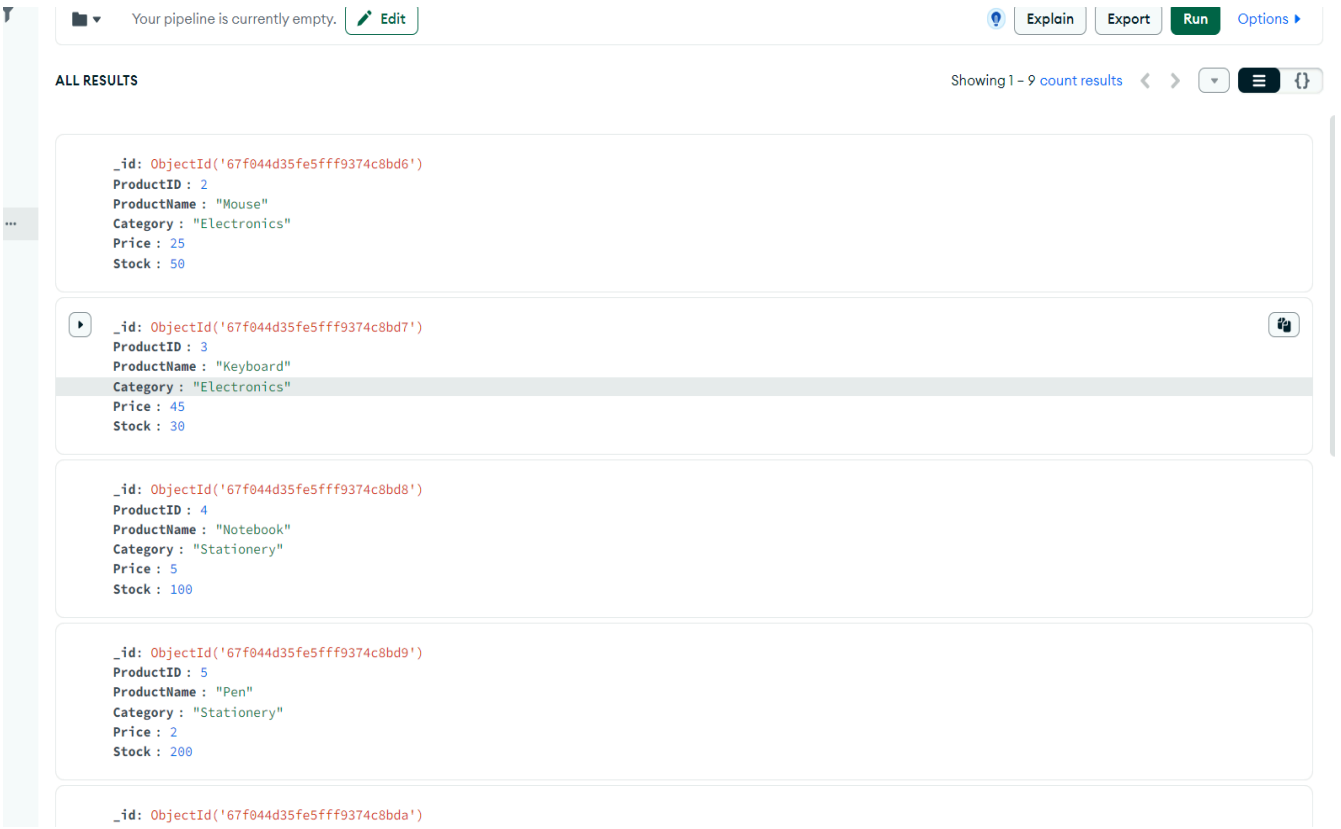
Display the Number of Products in the "Electronics" Category



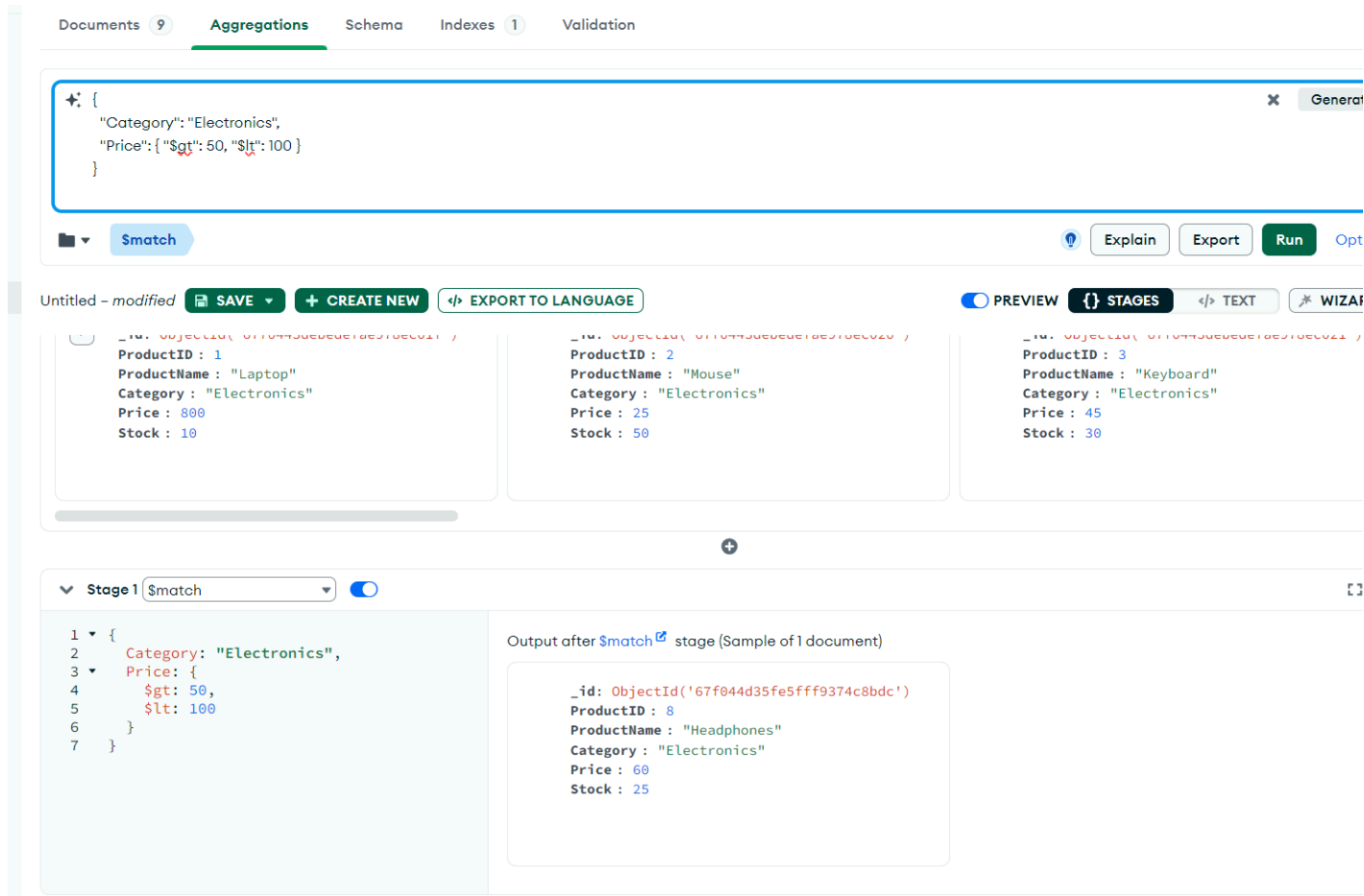
Display All Products Without Showing the "_id" Field



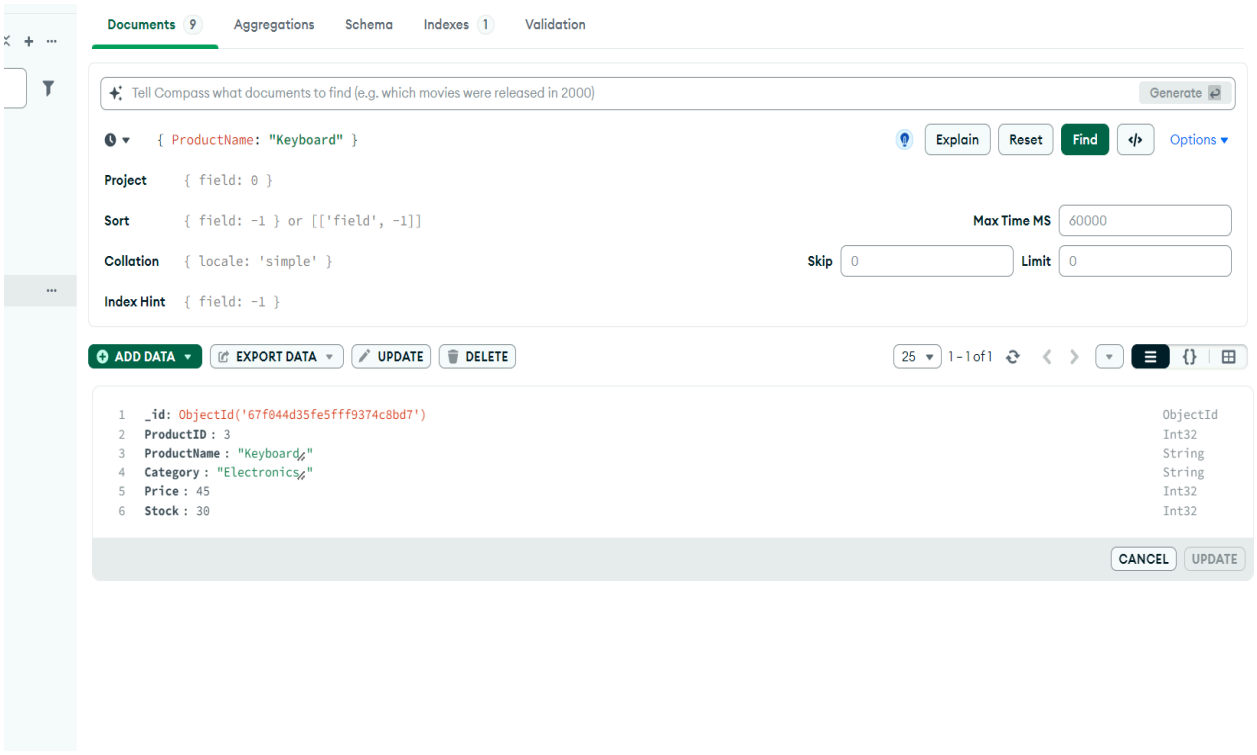
Display All Distinct Categories

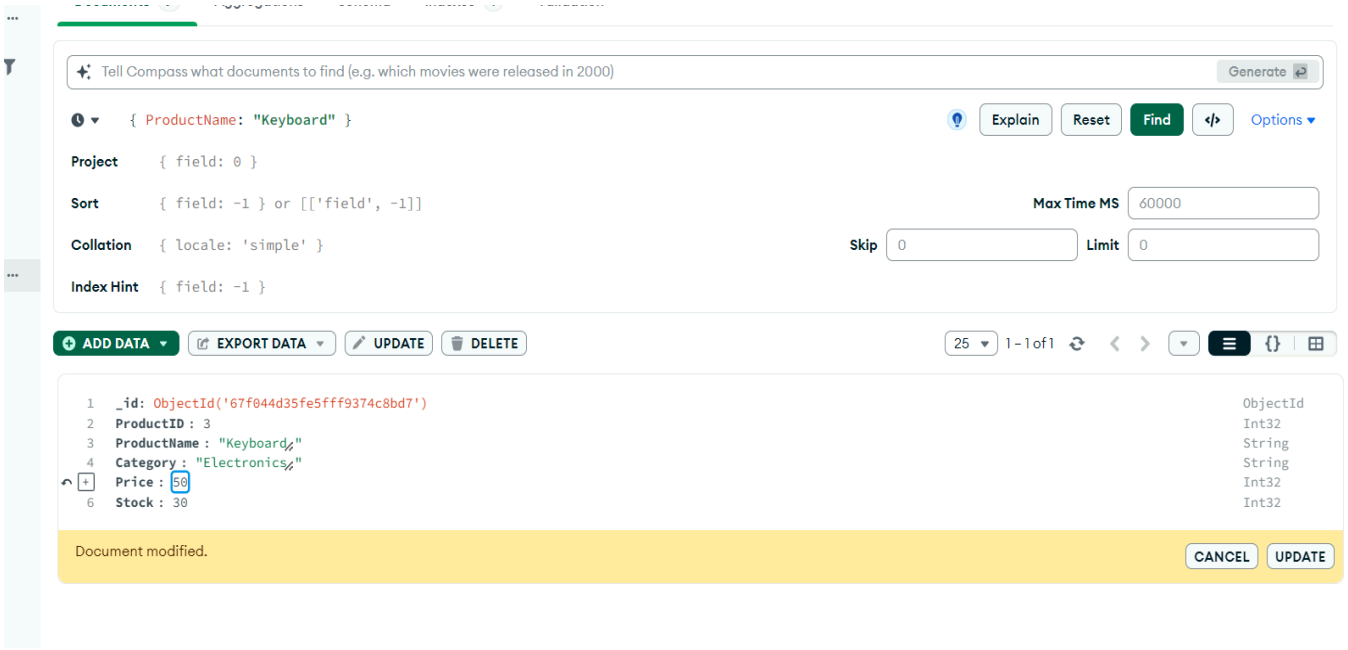


Display Products in "Electronics" With Prices Between 50 and 100

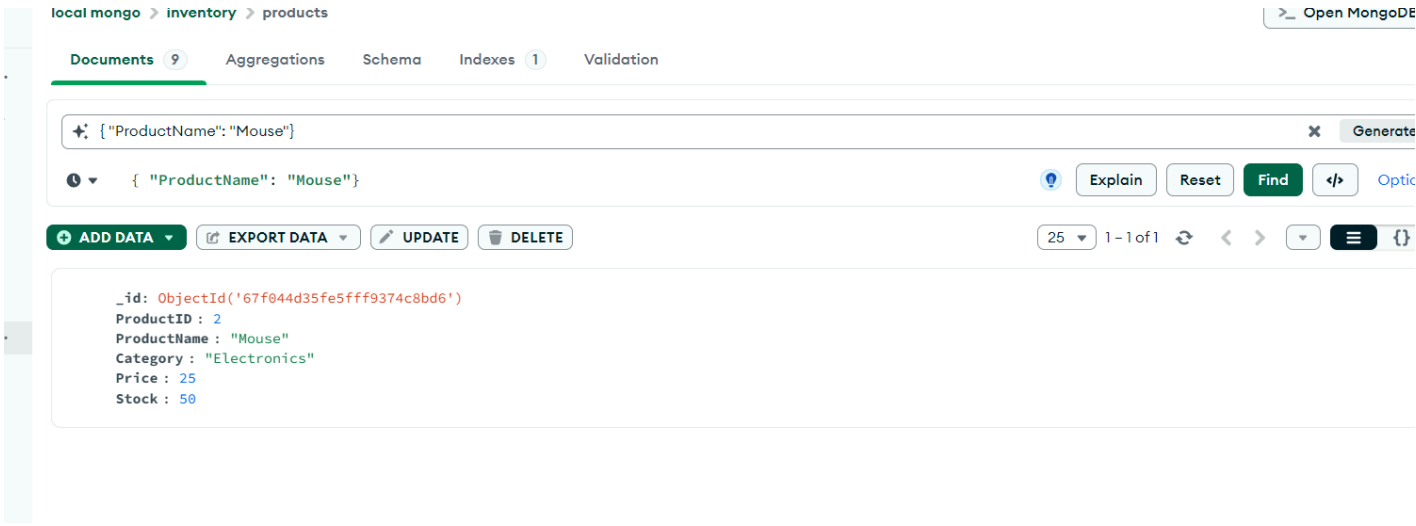


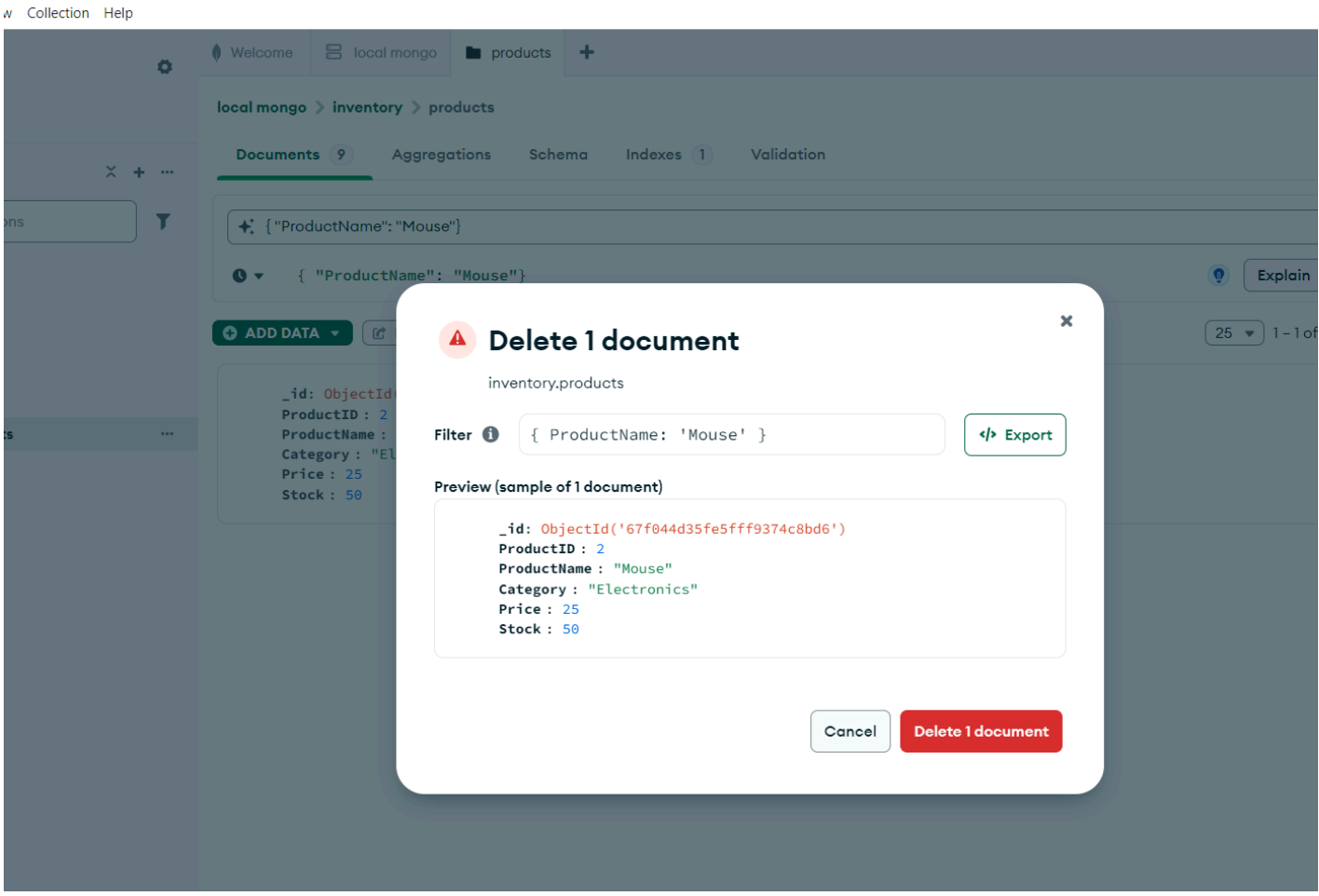
Change the Price of a Product (e.g., Update the Price of "Smartphone" to 650)





Delete a Particular Product Entry (e.g., Remove "Mouse")


local mongo > inventory > products
Open MongoDB
Documents 9
Aggregations
Schema
Indexes 1
Validation
{"ProductName": "Mouse"}
Generate
{ "ProductName": "Mouse"}
Explain
Reset
Find
ADD DATA
EXPORT DATA
UPDATE
DELETE
25
1 - 1 of 1
_id: ObjectId('67f044d35fe5fff9374c8bd6')
ProductID : 2
ProductName : "Mouse"
Category : "Electronics"
Price : 25
Stock : 50


w Collection Help
Welcome
local mongo
products
local mongo > inventory > products
Documents 9
Aggregations
Schema
Indexes 1
Validation
{"ProductName": "Mouse"}
Explain
ADD DATA
Delete 1 document
inventory.products
Filter
{ ProductName: 'Mouse' }
Export
Preview (sample of 1 document)
_id: ObjectId('67f044d35fe5fff9374c8bd6')
ProductID : 2
ProductName : "Mouse"
Category : "Electronics"
Price : 25
Stock : 50
Cancel
Delete 1 document