

Experiment – 6: MongoDB

| | |
|-----------------|----------------|
| Name of Student | Prajwal Pandey |
| Class Roll No | D15A / 32 |
| 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:

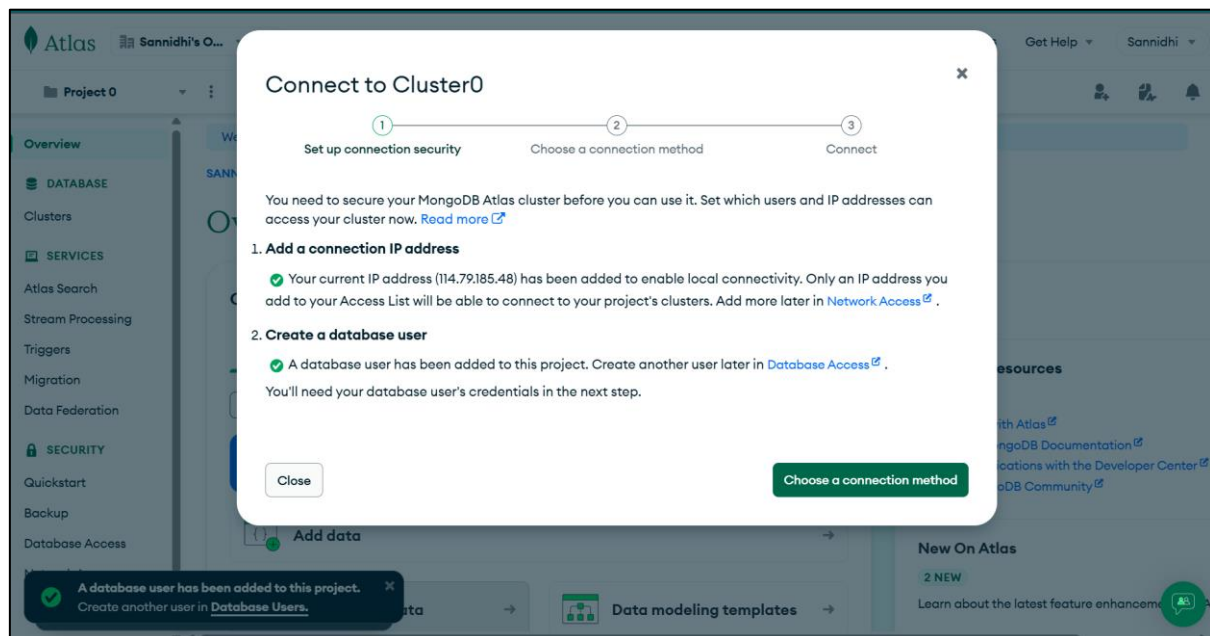
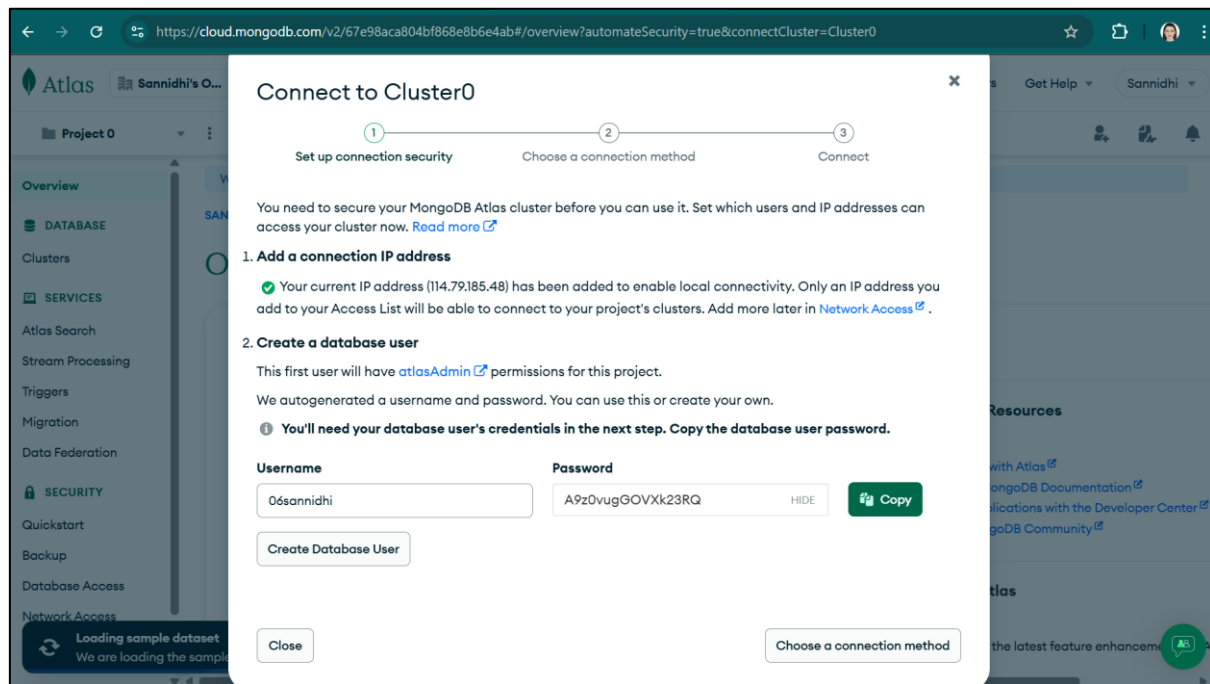
- Shards: Store actual data.
- Config Servers: Maintain metadata and sharding configuration.
- Mongos: Routes queries to the appropriate shards.

Benefits:

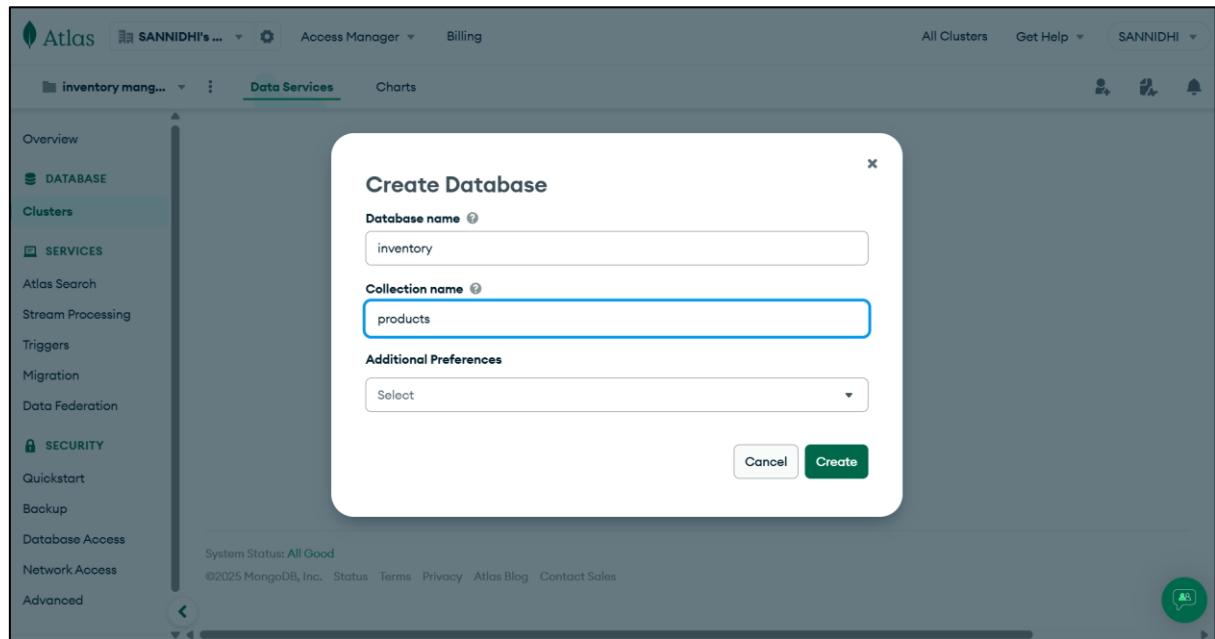
- Supports large-scale data management.
- Improves read and write performance.
- Ensures fault tolerance and high availability.

Output:

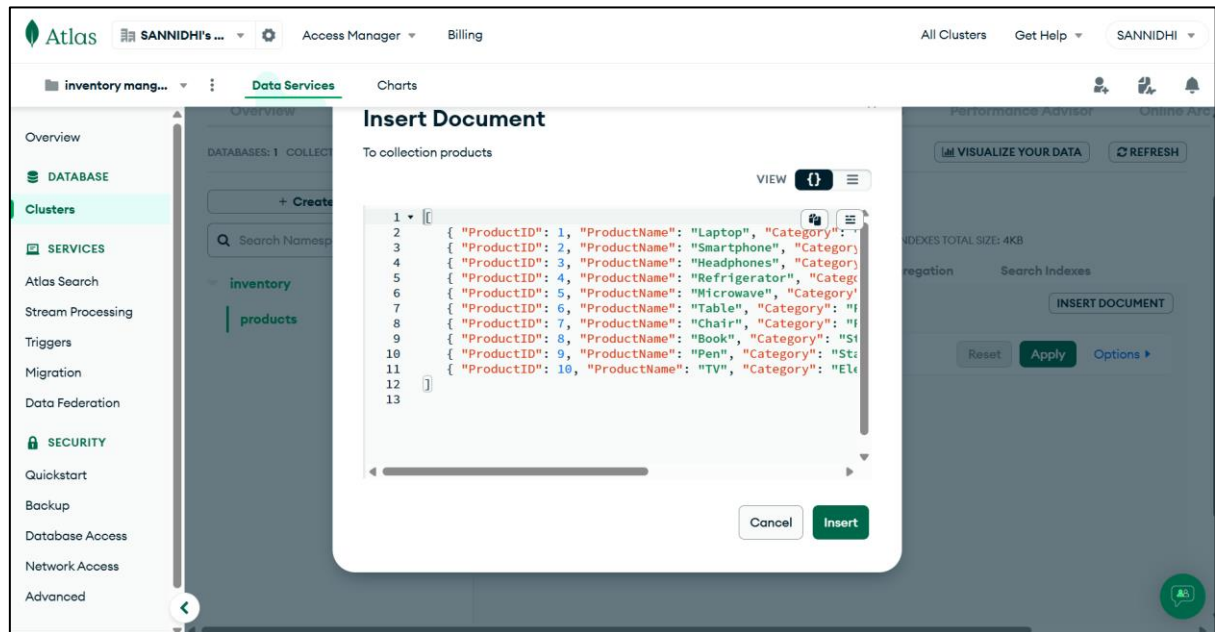
Connecting MongoDB Compass to Atlas Cluster



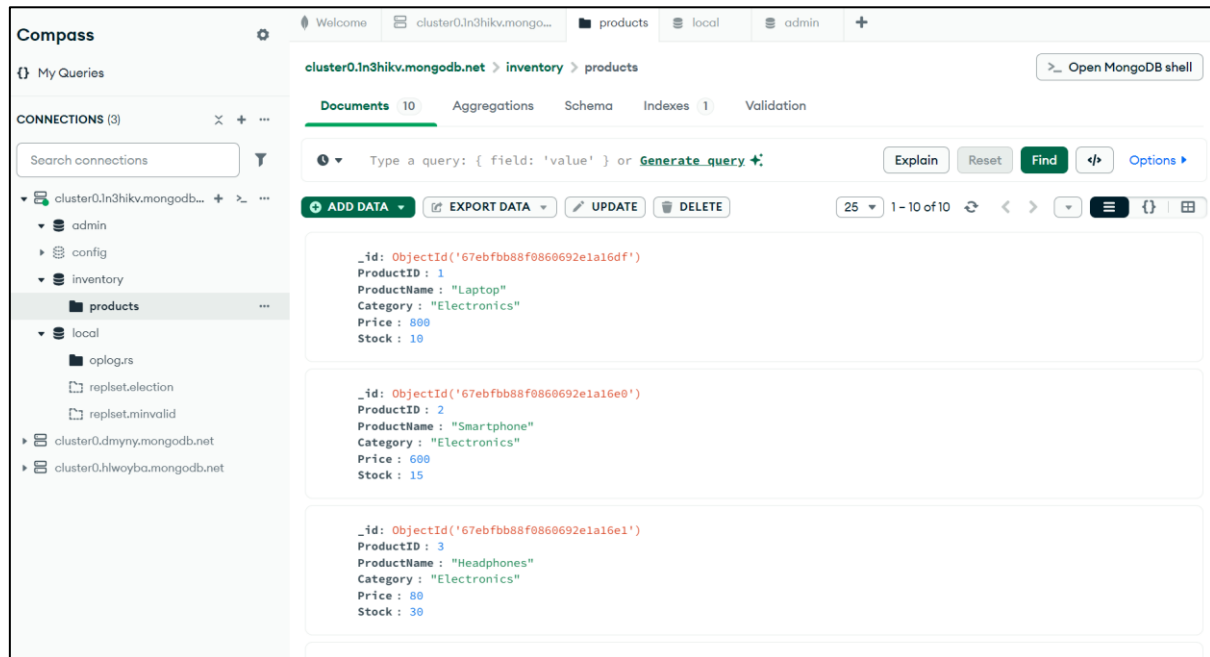
Create a Database and Collection



Insert 10 Documents into "products" Collection



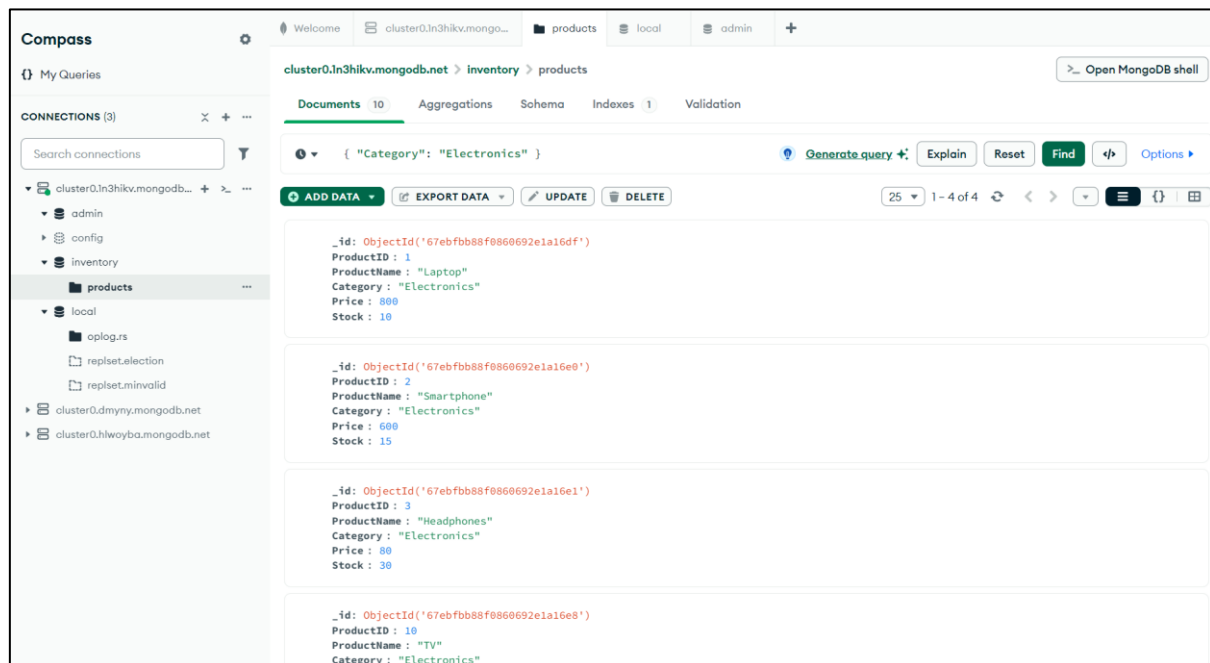
Display All Documents in "products" Collection



The screenshot shows the MongoDB Compass interface. The left sidebar displays the 'CONNECTIONS (3)' list with 'cluster0.in3hikv.mongodb.net' selected. The main panel shows the 'products' collection with 10 documents. The 'Documents' tab is active, displaying a list of documents. The query bar shows the default query: `{ field: 'value' }`. The document list shows three documents:

| ProductID | ProductName | Category | Price | Stock |
|-----------|-------------|-------------|-------|-------|
| 1 | Laptop | Electronics | 800 | 10 |
| 2 | Smartphone | Electronics | 600 | 15 |
| 3 | Headphones | Electronics | 80 | 30 |

Display All Products in the "Electronics" Category



The screenshot shows the MongoDB Compass interface with a query applied to filter documents by category. The query bar shows the query: `{ "Category": "Electronics" }`. The document list shows four documents:

| ProductID | ProductName | Category | Price | Stock |
|-----------|-------------|-------------|-------|-------|
| 1 | Laptop | Electronics | 800 | 10 |
| 2 | Smartphone | Electronics | 600 | 15 |
| 3 | Headphones | Electronics | 80 | 30 |
| 10 | TV | Electronics | | |

Display All Products in Ascending Order of Their Names

The screenshot shows the MongoDB Compass interface. The left sidebar displays the 'CONNECTIONS' list with 'cluster0.in3hkv.mongodb.net' selected. The main panel shows the 'products' collection in the 'inventory' database. The 'Documents' tab is active, displaying a list of products. The query editor shows the following query:

```
{}
```

The query options are set to 'Project' (field: 0), 'Sort' ('ProductName': 1), 'Collation' (locale: 'simple'), and 'Index Hint' (field: -1). The 'Max Time MS' is set to 60000. The 'Skip' and 'Limit' fields are both set to 0. The results show four products in ascending order of their names:

- ProductID: 6, ProductName: "Book", Category: "Stationery", Price: 20, Stock: 50
- ProductID: 7, ProductName: "Chair", Category: "Furniture", Price: 75, Stock: 20
- ProductID: 3, ProductName: "Headphones", Category: "Electronics", Price: 80, Stock: 30
- ProductID: 1, ProductName: "Laptop", Category: "Electronics", Price: 800, Stock: 10

Display the First 5 Products

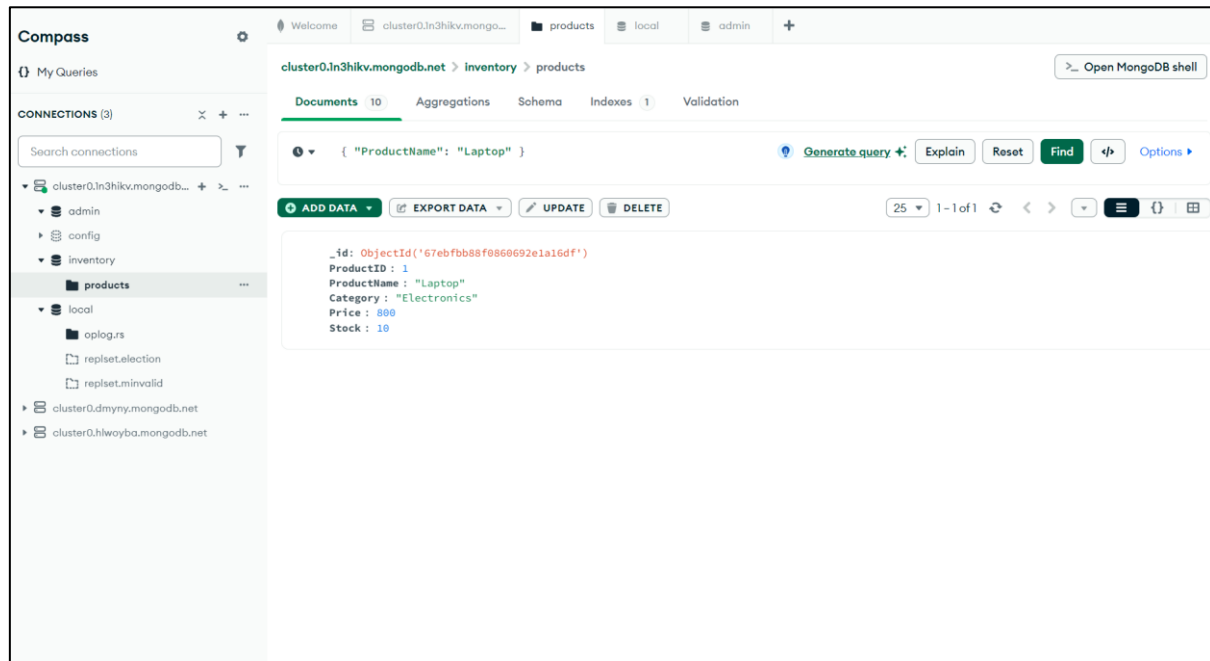
The screenshot shows the MongoDB Compass interface. The left sidebar displays the 'CONNECTIONS' list with 'cluster0.in3hkv.mongodb.net' selected. The main panel shows the 'products' collection in the 'inventory' database. The 'Documents' tab is active, displaying a list of products. The query editor shows the following query:

```
{}
```

The query options are set to 'Project' (field: 0), 'Sort' (field: -1) or [['field', -1]], 'Collation' (locale: 'simple'), and 'Index Hint' (field: -1). The 'Max Time MS' is set to 60000. The 'Skip' field is set to 0 and the 'Limit' field is set to 5. The results show the first 5 products in ascending order of their names:

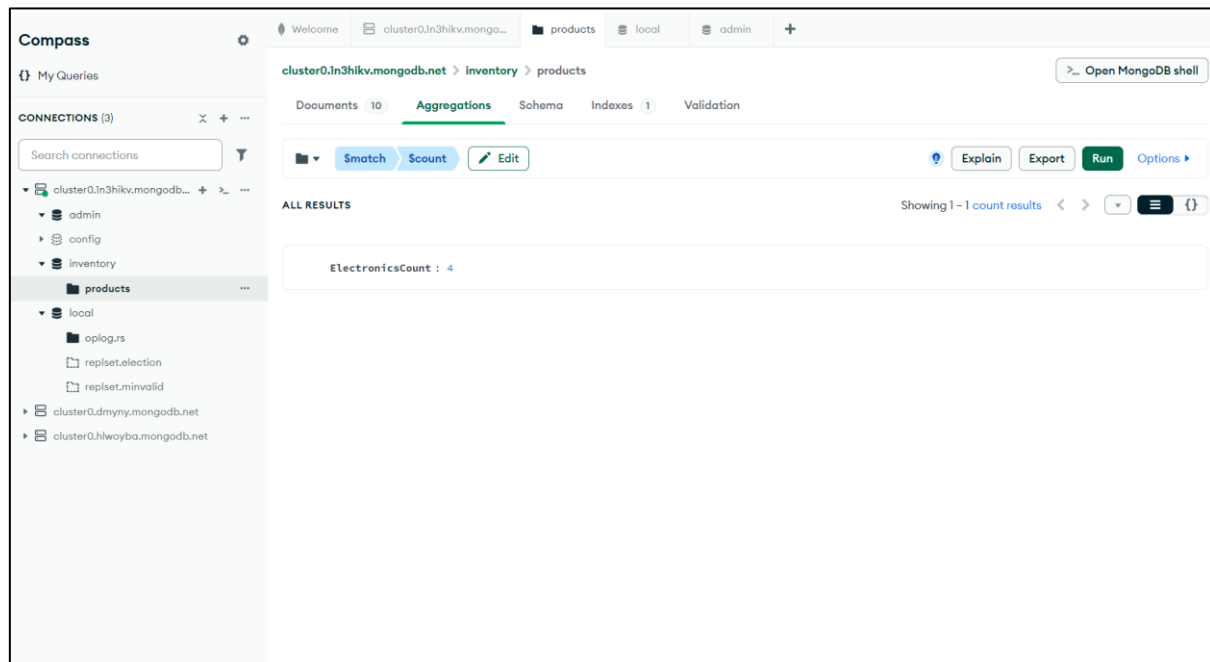
- ProductID: 1, ProductName: "Laptop", Category: "Electronics", Price: 800, Stock: 10
- ProductID: 2, ProductName: "Smartphone", Category: "Electronics", Price: 600, Stock: 15
- ProductID: 3, ProductName: "Headphones", Category: "Electronics", Price: 80, Stock: 30
- ProductID: 4, ProductName: "Refrigerator", Category: "Appliances", Price: 1200, Stock: 10
- ProductID: 6, ProductName: "Book", Category: "Stationery", Price: 20, Stock: 50

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



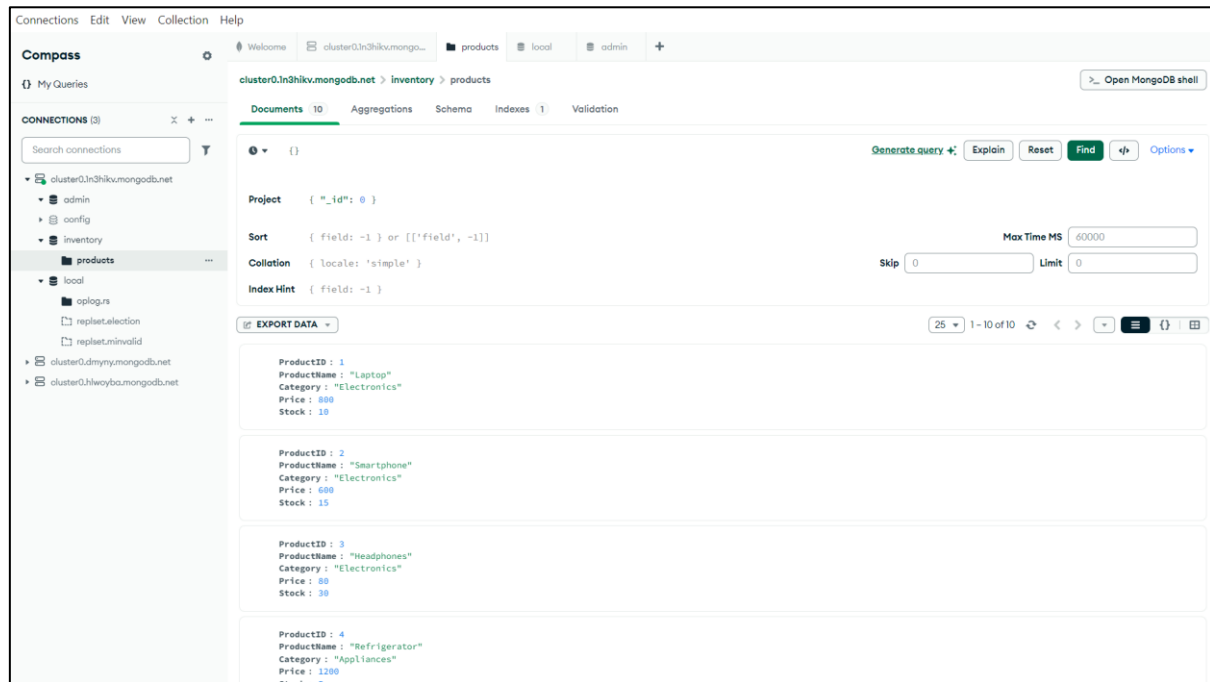
The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS (3)' panel lists several clusters, with 'cluster0.in3hikv.mongodb.net' selected. The main panel displays the 'products' collection under the 'inventory' database. The 'Documents' tab is active, showing a single document with the following fields: `_id`: ObjectId('67ebfbb88f0860692e1a16df'), `ProductID`: 1, `ProductName`: "Laptop", `Category`: "Electronics", `Price`: 800, and `Stock`: 10. The query bar at the top contains the filter `{ "ProductName": "Laptop" }`. Below the query bar, there are buttons for 'ADD DATA', 'EXPORT DATA', 'UPDATE', and 'DELETE'. The document is displayed in a JSON format.

Display the Number of Products in the "Electronics" Category



The screenshot shows the MongoDB Compass interface. On the left, the 'CONNECTIONS (3)' panel lists several clusters, with 'cluster0.in3hikv.mongodb.net' selected. The main panel displays the 'products' collection under the 'inventory' database. The 'Aggregations' tab is active, showing a single aggregation pipeline with the `$match` stage. The query bar at the top contains the filter `{ "Category": "Electronics" }`. Below the query bar, there are buttons for 'Explain', 'Export', 'Run', and 'Options'. The 'ALL RESULTS' section shows the output of the aggregation: `ElectronicsCount : 4`. The results are displayed in a JSON format.

Display All Products Without Showing the "_id" Field



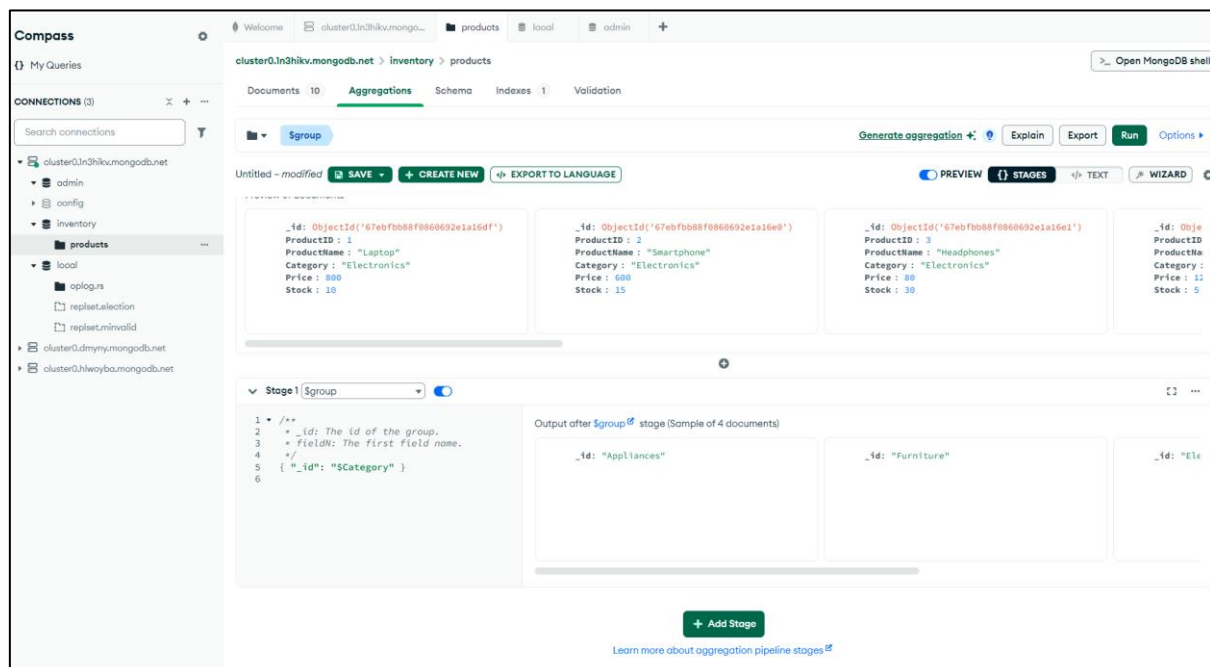
The screenshot shows the MongoDB Compass interface. The left sidebar displays the 'products' collection. The main panel shows a query editor with the following configuration:

- Project:** { "_id": 0 }
- Sort:** { field: -1 } or [{ 'field', -1 }]
- Collation:** { locale: 'simple' }
- Index Hint:** { field: -1 }

The 'EXPORT DATA' tab is selected, showing a table of product data:

| ProductID | ProductName | Category | Price | Stock |
|-----------|--------------|-------------|-------|-------|
| 1 | Laptop | Electronics | 800 | 10 |
| 2 | Smartphone | Electronics | 600 | 15 |
| 3 | Headphones | Electronics | 80 | 30 |
| 4 | Refrigerator | Appliances | 1200 | 5 |

Display All Distinct Categories



The screenshot shows the MongoDB Compass interface with an aggregation pipeline configured to display all distinct categories. The 'Aggregations' tab is selected, and the pipeline is named 'Untitled - modified'.

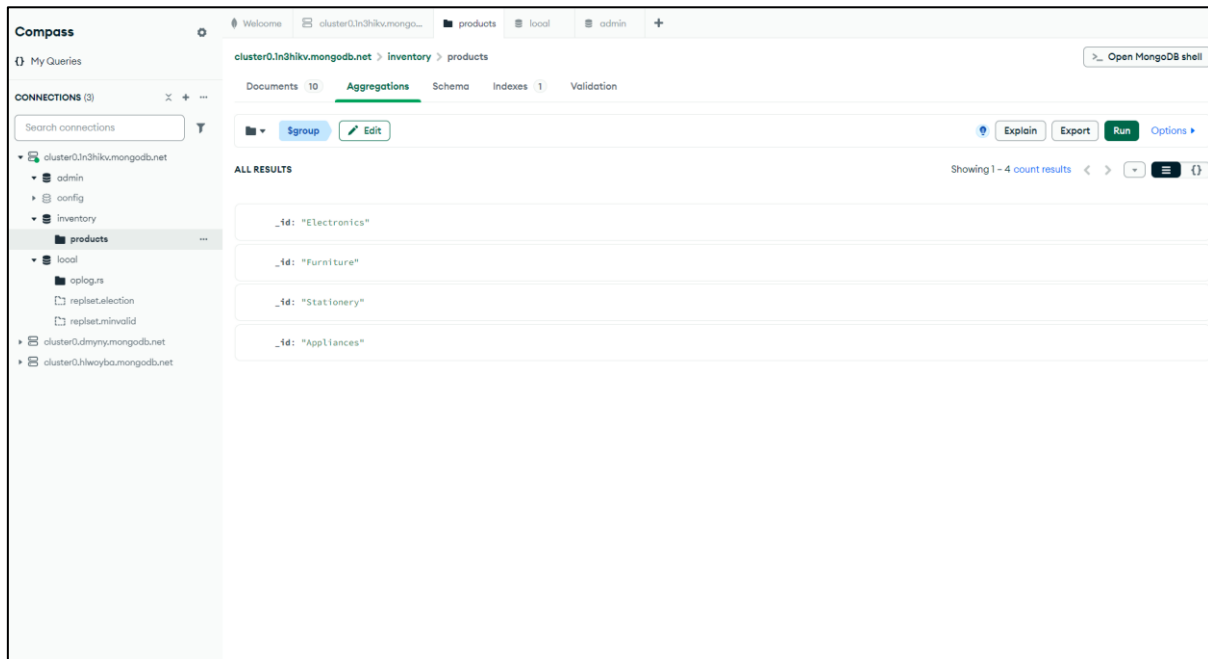
Stage 1: \$group

```
1 /*  
2 * _id: The id of the group.  
3 * fieldN: The first field name.  
4 */  
5 { "_id": "$Category" }  
6
```

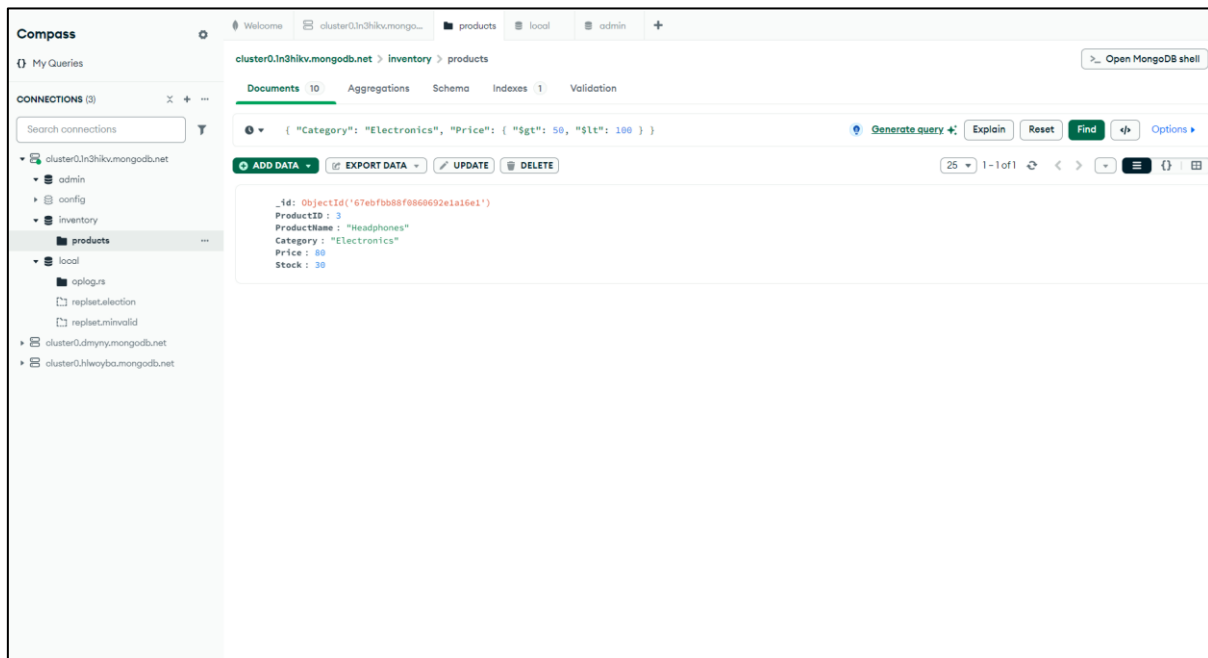
The 'Preview' tab is selected, showing the output of the aggregation pipeline (Sample of 4 documents):

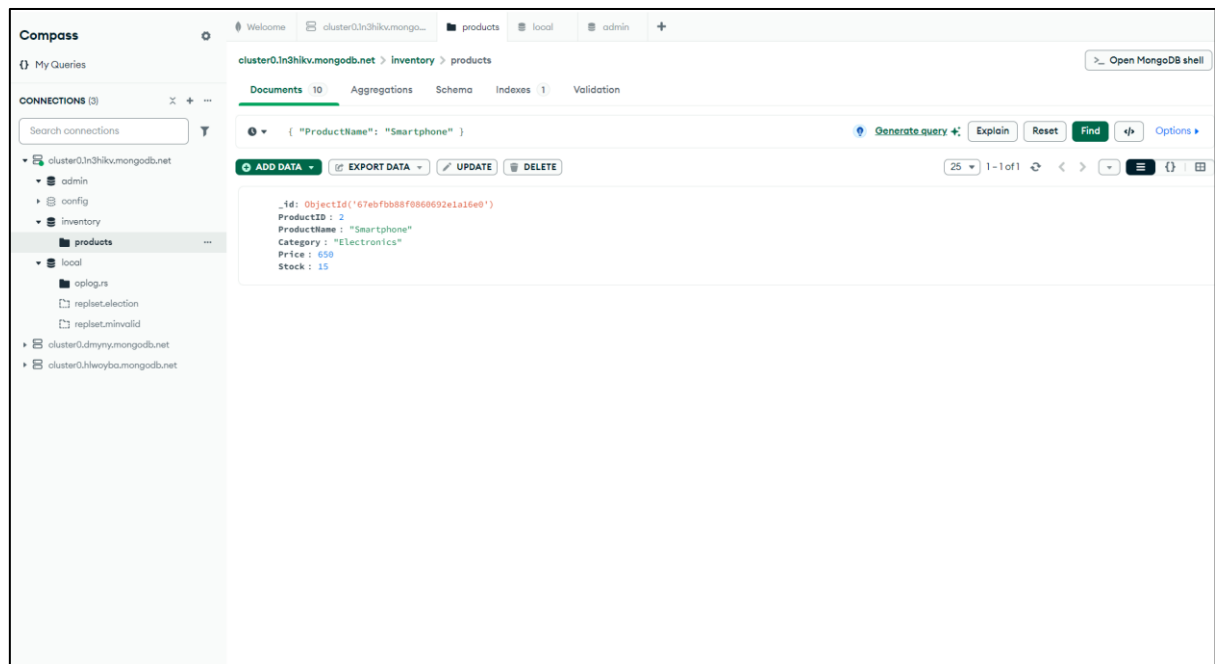
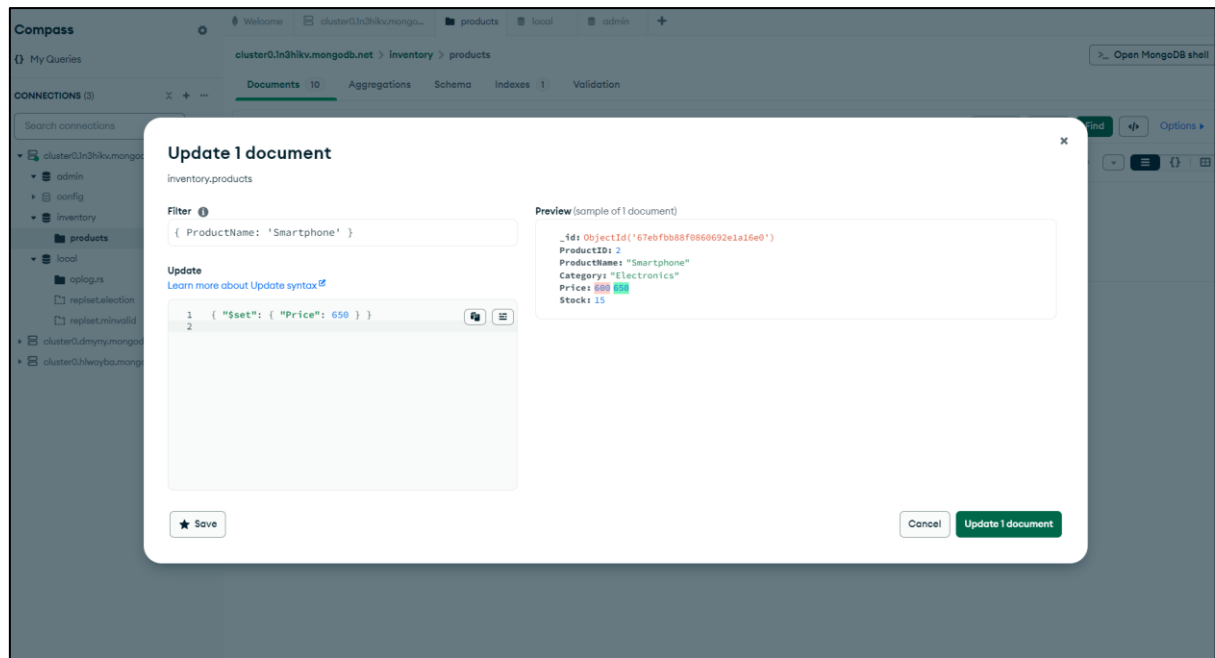
| _id |
|---------------|
| "Appliances" |
| "Furniture" |
| "Electronics" |

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 "Pen")

