Experiment – 6: MongoDB

Name of Student	Spandan Deb
Class Roll No	13
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. Features of MongoDB

MongoDB is a NoSQL database that provides high performance, scalability, and flexibility. Some of its key features include:

1. Document-Oriented Storage: Stores data in JSON-like BSON format, making it flexible and easy to query.
2. Schema-Less Database: No predefined schema is required, allowing for dynamic and evolving data structures.
3. Scalability: Supports horizontal scaling through sharding to distribute data across multiple servers.
4. Replication: Provides high availability with automatic failover using replica sets.
5. Indexing: Supports various types of indexes to improve query performance.
6. Aggregation Framework: Allows powerful data transformation and computation within the database.
7. High Performance: Efficient for read and write operations, especially for big data applications.
8. Support for Geospatial Queries: Enables location-based data storage and retrieval.
9. Flexible Query Language: Uses rich queries with filtering, sorting, and projection capabilities.

B. Documents and Collections in MongoDB

Documents

- A document in MongoDB is a record stored in BSON (Binary JSON) format.
- It is similar to a row in a relational database but is more flexible.
- Documents contain key-value pairs and can have nested structures.

Example of a MongoDB Document:

```
{
  "_id": ObjectId("507f1f77bcf86cd799439011"),
  "name": "John Doe",
  "age": 30,
  "address": {
    "city": "New York",
    "zip": "10001"
```

```
    }
}
```

Collections

- A collection is a group of related documents in MongoDB, similar to a table in a relational database.
- Collections do not require a fixed schema, allowing documents to have different structures.

C. When to Use MongoDB?

MongoDB is suitable for various use cases, including:

1. Big Data Applications: Efficiently handles large volumes of unstructured or semi-structured data.
2. Real-Time Analytics: Supports fast read/write operations for real-time data processing.
3. Content Management Systems (CMS): Flexible schema allows storing diverse types of content.
4. Internet of Things (IoT)**:** Handles high-velocity data ingestion from IoT devices.
5. E-Commerce Applications: Provides dynamic and scalable product catalogs.
6. Mobile and Web Applications: Works well with APIs and microservices architectures.
7. Social Media Applications: Supports large-scale user-generated content with fast retrieval.

D. What is Sharding in MongoDB?

Definition

Sharding is a method used in MongoDB to distribute large datasets across multiple servers to ensure high availability and scalability**.**

How Sharding Works?

1. Shard**:** A single database instance that holds a portion of the data.
2. Config Server**:** Stores metadata and configuration settings for the cluster.

3. Query Router (Mongos)**:** Directs client requests to the appropriate shard based on the shard key.

Sharding Process:

- Data is partitioned into smaller pieces called chunks based on a shard key**.**
- Chunks are distributed across multiple shards.
- When queries are made, the query router determines which shard contains the relevant data and directs the request accordingly.

Benefits of Sharding:

- Horizontal Scalability: Spreads data across multiple machines, reducing load on a single server.
- Improved Performance: Parallel processing of queries across multiple shards.
- High Availability**:** If one shard fails, data can still be accessed from others.

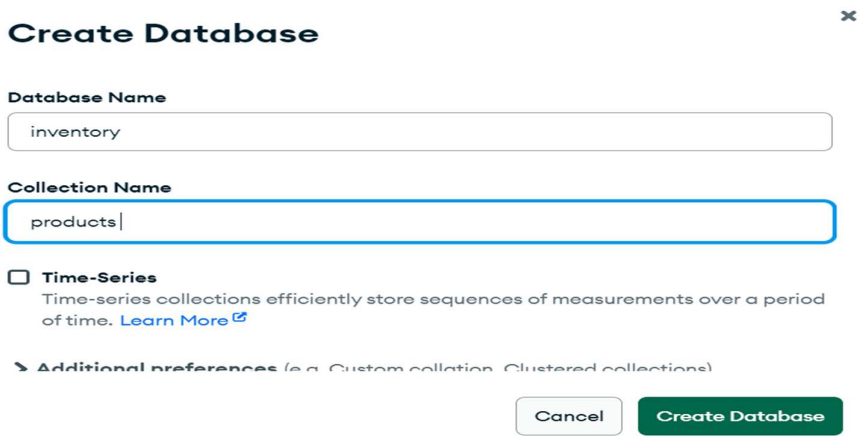
Example of Sharding in MongoDB:

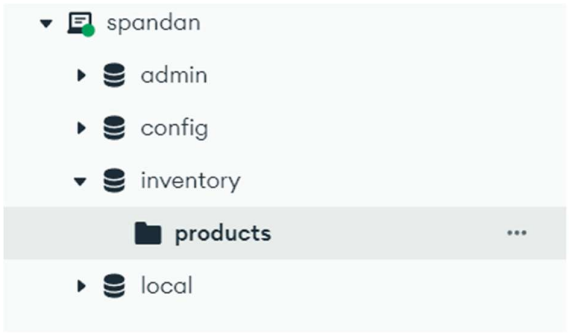
```
sh.enableSharding("myDatabase")
sh.shardCollection("myDatabase.myCollection", { "userId": 1 })
```

This enables sharding for myDatabase and distributes myCollection based on userId.

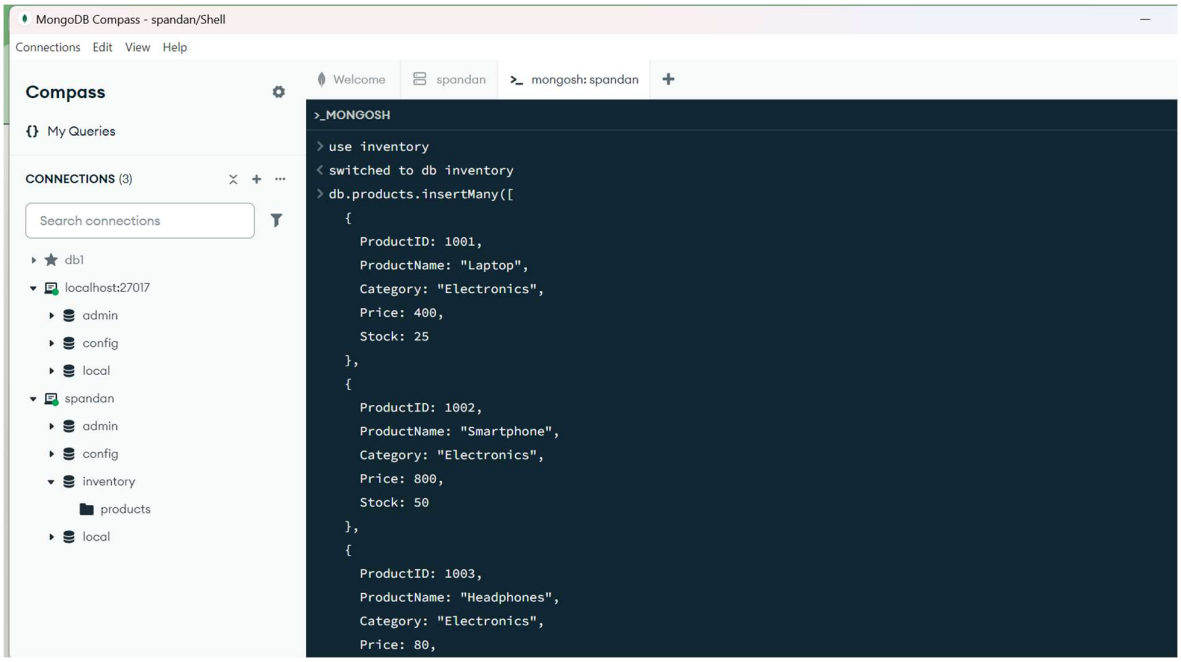
OUTPUT:

Created a database inventory and collection products





Insert 10 Documents in Products Collections



```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67ea9783f291a474bf0c8b63'),
    '1': ObjectId('67ea9783f291a474bf0c8b64'),
    '2': ObjectId('67ea9783f291a474bf0c8b65'),
    '3': ObjectId('67ea9783f291a474bf0c8b66'),
    '4': ObjectId('67ea9783f291a474bf0c8b67'),
    '5': ObjectId('67ea9783f291a474bf0c8b68'),
    '6': ObjectId('67ea9783f291a474bf0c8b69'),
    '7': ObjectId('67ea9783f291a474bf0c8b6a'),
    '8': ObjectId('67ea9783f291a474bf0c8b6b'),
    '9': ObjectId('67ea9783f291a474bf0c8b6c')
  }
}
inventory>
```

Display all the documents in the products collection

```
> db.products.find()
{
  _id: ObjectId('67ea9783f291a474bf0c8b63'),
  ProductID: 1001,
  ProductName: 'Laptop',
  Category: 'Electronics',
  Price: 400,
  Stock: 25
}
{
  _id: ObjectId('67ea9783f291a474bf0c8b64'),
  ProductID: 1002,
  ProductName: 'Smartphone',
  Category: 'Electronics',
  Price: 800,
  Stock: 50
}
{
  _id: ObjectId('67ea9783f291a474bf0c8b65'),
```

Display all products in electronics category

```
> db.products.find({Category:'Electronics'})
< {
    _id: ObjectId('67ea9783f291a474bf0c8b63'),
    ProductID: 1001,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 400,
    Stock: 25
  }
  {
    _id: ObjectId('67ea9783f291a474bf0c8b64'),
    ProductID: 1002,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 800,
    Stock: 50
  }
  {
    _id: ObjectId('67ea9783f291a474bf0c8b65'),
    ProductID: 1003,
    ProductName: 'Headphones',
    Category: 'Electronics',
```

Display all the products in ascending order of their name

```
> db.products.find().sort({ ProductName: 1 })
< {
    _id: ObjectId('67ea9783f291a474bf0c8b69'),
    ProductID: 1007,
    ProductName: 'Bluetooth Speaker',
    Category: 'Electronics',
    Price: 49.99,
    Stock: 75
  }
  {
    _id: ObjectId('67ea9783f291a474bf0c8b6b'),
    ProductID: 1009,
    ProductName: 'Bookshelf',
    Category: 'Furniture',
    Price: 89.99,
    Stock: 10
  }
  {
    _id: ObjectId('67ea9783f291a474bf0c8b68'),
    ProductID: 1006,
    ProductName: 'Coffee Maker',
```

Display the details of first 5 products

```
> db.products.find().limit(5)
< {
    _id: ObjectId('67ea9783f291a474bf0c8b63'),
    ProductID: 1001,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 400,
    Stock: 25
  }
  {
    _id: ObjectId('67ea9783f291a474bf0c8b64'),
    ProductID: 1002,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 800,
    Stock: 50
  }
  {
    _id: ObjectId('67ea9783f291a474bf0c8b65'),
    ProductID: 1003,
    ProductName: 'Headphones',
    Category: 'Electronics',
```

Display the category of product

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

Display the number of products in electronics category

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

Display all the products without _id field

```
> db.products.find({}, { _id: 0 })
< {
    ProductID: 1001,
    ProductName: 'Laptop',
    Category: 'Electronics',
    Price: 400,
    Stock: 25
  }
  {
    ProductID: 1002,
    ProductName: 'Smartphone',
    Category: 'Electronics',
    Price: 800,
    Stock: 50
  }
```

Display all the distinct categories of products

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

Display products in the electronics category with price greater than 50 and less than 100

```
> db.products.find({
    Category: "Electronics",
    Price: { $gt: 50, $lt: 100 }
  })
< {
    _id: ObjectId('67ea9783f291a474bf0c8b65'),
    ProductID: 1003,
    ProductName: 'Headphones',
    Category: 'Electronics',
    Price: 80,
    Stock: 100
  }
```

Change the price of a product (eg Update laptop price to 1300)

```
> db.products.updateOne(
    { ProductName: "Laptop" },
    { $set: { Price: 1300 } }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
```

Delete a specific product

```
> db.products.deleteOne({ ProductID: 1005 })
< {
    acknowledged: true,
    deletedCount: 1
  }
```

CONCLUSION:

MongoDB is a powerful NoSQL database offering flexible document storage, scalability through sharding, and efficient querying. It is ideal for handling large datasets, real-time applications, and dynamic content management. Sharding in MongoDB enables horizontal scaling, improving both performance and reliability in distributed environments.