**St. Francis Institute of Technology Borivali (West), Mumbai-400103**

**(Autonomous Institute)**

**Department of Information Technology**

**Academic Year: 2024-25**

**Class: TE-ITA/B                                                                            Semester: VI**

**Subject: Web Lab**

**Experiment – 6: Install MongoDB, perform CRUD operations using the Mongo shell, and utilize MongoDB Compass to query and analyze data effectively, Install and use MongoDB Compass to query and analyze MongoDB data**

1. **Aim:** Installation of MongoDB and perform CRUD operations using Mongo shell, Install and use MongoDB Compass to query and analyze MongoDB data.
2. **Objectives:** Aim of this experiment is that, the students will be able

* To install MongoDB and MongoDB compass
* Read and understand commonly-used MongoDB syntax
* Environment setup
* To understand the use of tools available in compass to query MongoDB data.

1. **Outcomes:** After study of this experiment, the students will be able

* To write and execute queries for performing CRUD operations in MongoDB
* To create and manage databases using MongoDB Compass GUI.
* To handle coding and syntax error

1. **Prerequisite:** Basic understanding of database,SQL and OOPs, text editor and execution of programs, Basic understanding of database, MongoDB commands, data types
2. **Requirements:** Personal Computer, Windows operating system, MongoDB, Mongo shell browser, Internet Connection, google doc.
3. **Pre-Experiment Exercise:**

**Brief Theory:** Refer shared material

1. **Laboratory Exercise**
   * + 1. **Procedure:**

**a. Answer the following:**

## **What are some of the advantages of MongoDB?**

**ANS:**

1. **Flexible Schema**: MongoDB uses a schema-less structure, allowing for flexible data storage and enabling changes to data models without downtime.
2. **Scalability**: MongoDB supports horizontal scaling through sharding, which allows data to be distributed across multiple machines.
3. **High Availability**: With replica sets, MongoDB provides automatic failover and data redundancy, ensuring high availability of data.
4. **Document-Oriented**: MongoDB stores data in BSON (Binary JSON) format, which is more intuitive and allows complex data structures, such as arrays and nested objects.
5. **Powerful Querying**: MongoDB supports a wide range of queries, including text search, geospatial queries, and aggregation operations.
6. **Performance**: Its indexing and in-memory processing features enable fast query execution, even with large datasets.
7. **Open Source**: MongoDB is open-source and offers a wide range of community support and resources.

## **2. What is a Document in MongoDB?**

**ANS:**  A **Document** in MongoDB is a basic unit of data, similar to a row in relational databases, but it is more flexible. It is represented in BSON (Binary JSON) format, allowing for embedded arrays and objects. Documents can store various data types, including strings, integers, dates, arrays, and even other documents. A document in MongoDB typically contains a **unique identifier**, called the "\_id" field, which is used to reference the document.

eg. {

"\_id": ObjectId("6173c9a479e1c6a3ef2d94ed"),

"name": "John Doe",

"age": 30,

"address": {

"street": "123 Main St",

"city": "Anytown"

},

"hobbies": ["reading", "travelling"]

}

## **3. Explain Sharding in MongoDB.**

**ANS:**Sharding in MongoDB refers to the process of distributing data across multiple servers, called **shards**, to ensure horizontal scalability and improve performance when handling large datasets. When the data grows beyond the storage or processing capacity of a single machine, sharding allows MongoDB to scale out by adding more machines to the cluster.

Sharding is implemented by dividing the data into chunks based on a shard key, which is a field or set of fields in the documents. Each shard holds a subset of the data, and the **mongos** router directs queries to the appropriate shard. Sharding can improve query performance, increase storage capacity, and provide fault tolerance.

Key components of sharding:

* **Shard key**: The field used to distribute data across shards.
* **Chunks**: Subsets of data within a shard, created based on the shard key.
* **Mongos**: The routing service that directs client requests to the appropriate shard.
* **Config servers**: Store metadata about the sharded cluster.

## **4. What is the MongoDB Compass?**

**ANS**: **MongoDB Compass** is a graphical user interface (GUI) tool for MongoDB that allows users to interact with their MongoDB databases visually. It provides an easy-to-use interface for developers and database administrators to explore, analyze, and manage MongoDB data without writing complex queries.

Key features of MongoDB Compass:

1. **Visual Exploration**: It allows you to view and analyze your data in a more structured and visual way.
2. **Schema Visualization**: Compass can automatically analyze and visualize the schema of your collections, which helps in understanding the structure of the data.
3. **Query Builder**: It provides a graphical query builder for filtering and querying documents without having to write MongoDB queries manually.
4. **Aggregation Pipeline Builder**: You can build aggregation pipelines through a visual interface, making it easier to process and transform data.
5. **Performance Insights**: Compass gives performance metrics and insights, which help you identify slow queries and optimize them.

MongoDB Compass is available for Windows, Mac, and Linux, and it's especially useful for those who prefer working with a GUI over the command line.

## **5. How is the MongoDB compass different from the MongoDB shell?**

**ANS:**

| **MongoDB compass** | **MongoDB shell** |
| --- | --- |
| Graphical User Interface (GUI) | Command-Line Interface (CLI) |
| Beginner-friendly, intuitive GUI | Requires knowledge of MongoDB commands and JavaScript |
| Point-and-click, visual query builder, schema analysis | Command-based, flexible querying with full control |
| Visual query and aggregation builder | Manual query writing using JavaScript-based commands |
| Basic query performance and index analysis | Detailed system statistics, server status, and custom diagnostics |

**b**. **Attach screenshots:**

* MongoDB code and output with your own comments.
* MongoDB Compass screenshots.

1. **Post-Experiments Exercise**
2. **Extended Theory:**

Nil

1. **Questions:**

### What are some features of MongoDB?

### How to perform queries in MongoDB?

### What are the data types in MongoDB?

### Explain the three ways to access documents in MongoDB Compass.

### Explain the use of schema and aggregations tab available in Compass GUI.

1. **Conclusion:**

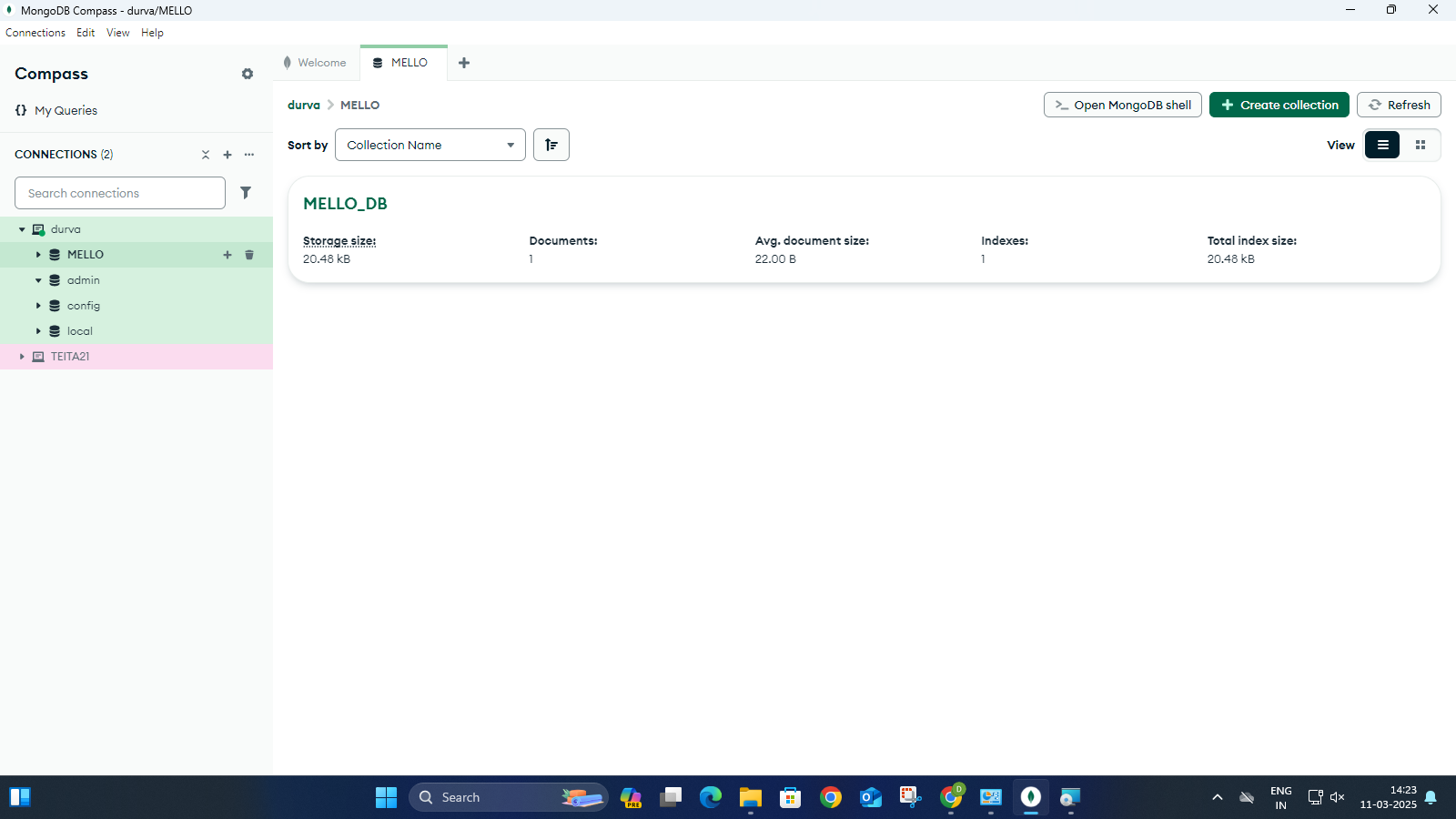
* Write what was performed in the experiment.
* Write the significance of the topic studied in the experiment.

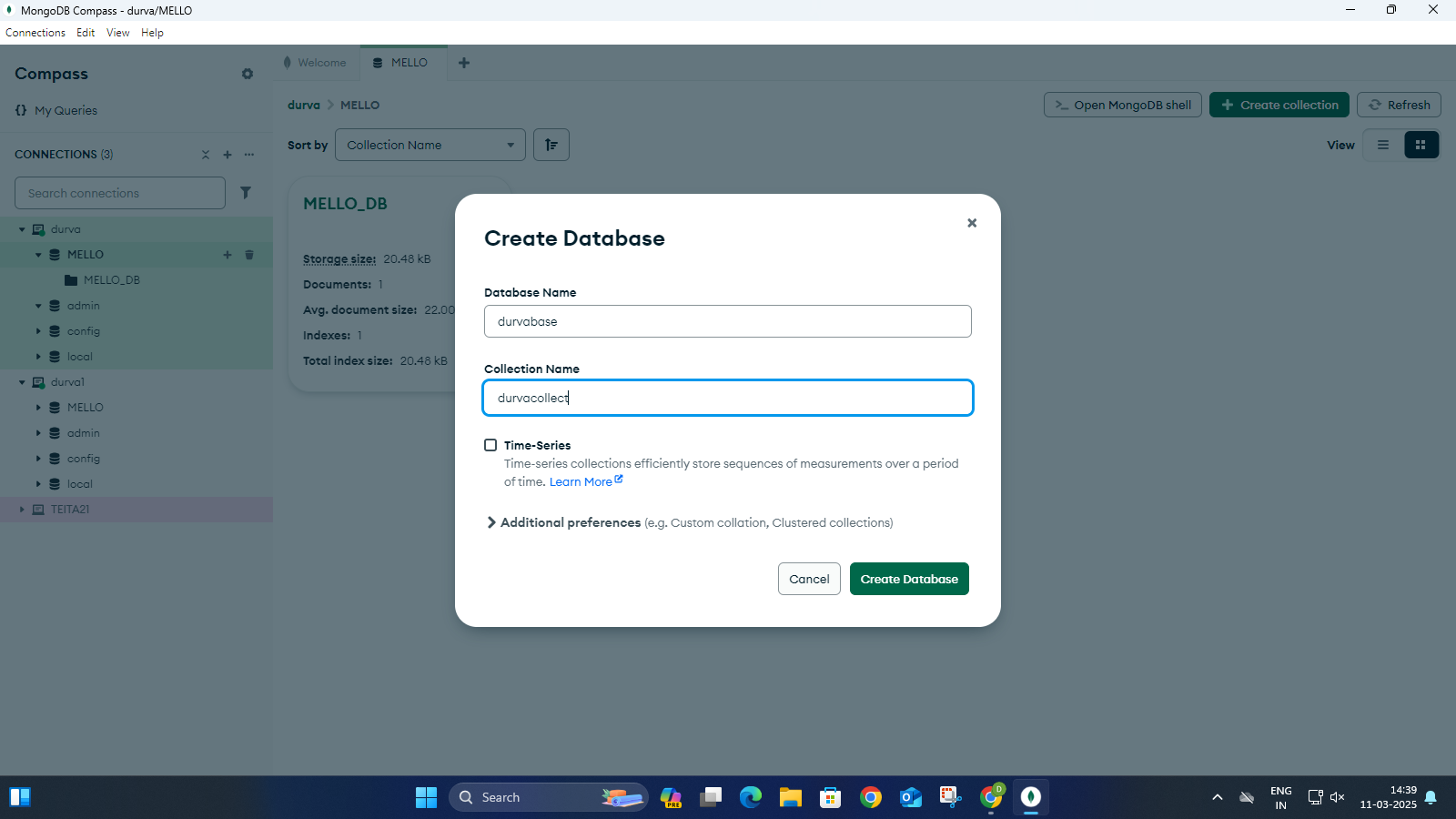
1. **References:**

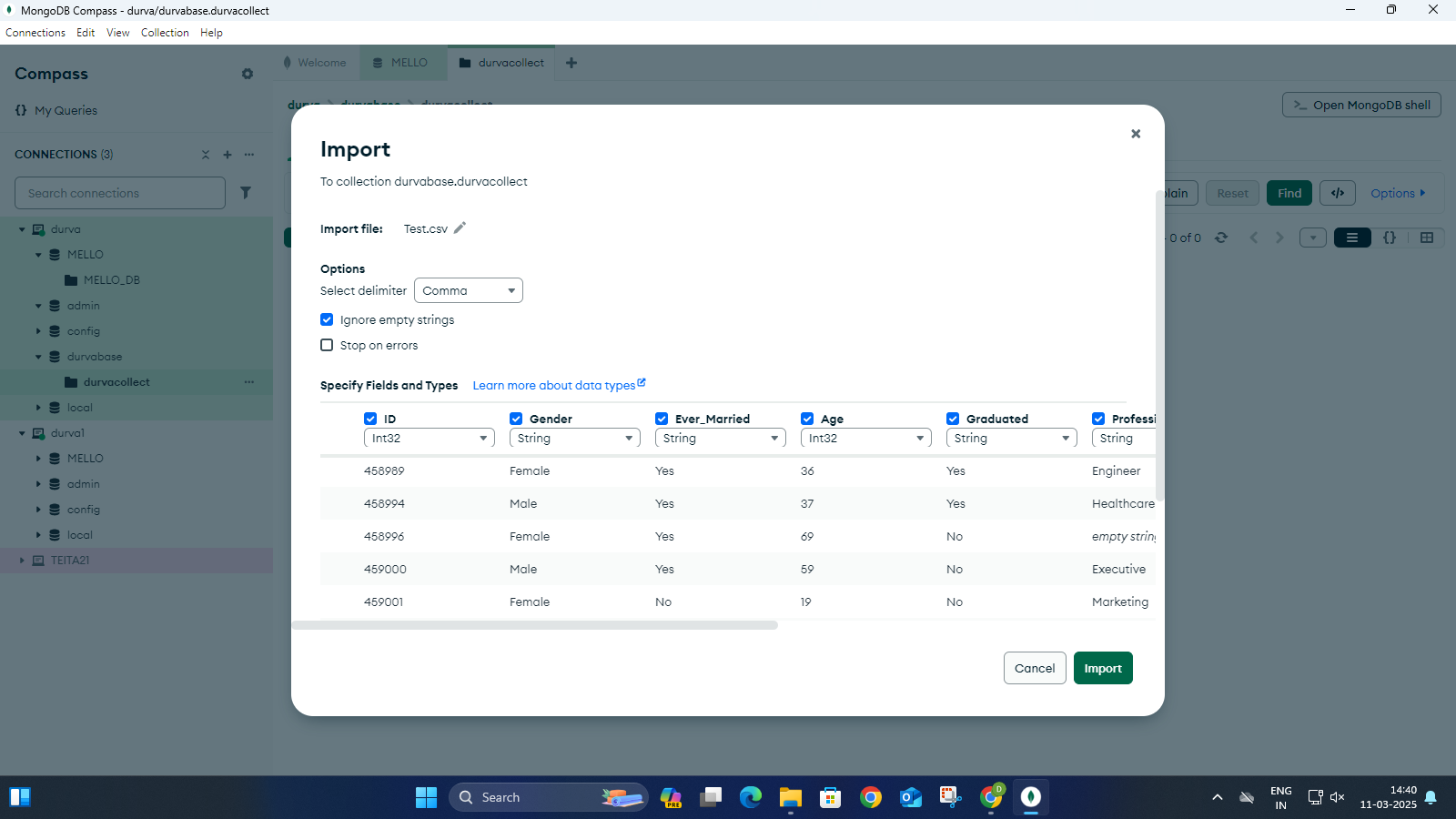
# MongoDB in Action, Second Edition[,](https://freepdf-books.com/beginning-javascript-with-dom-scripting-and-ajax-2nd-edition-pdf-free-download/) by Kyle Banker, Peter Bakkum

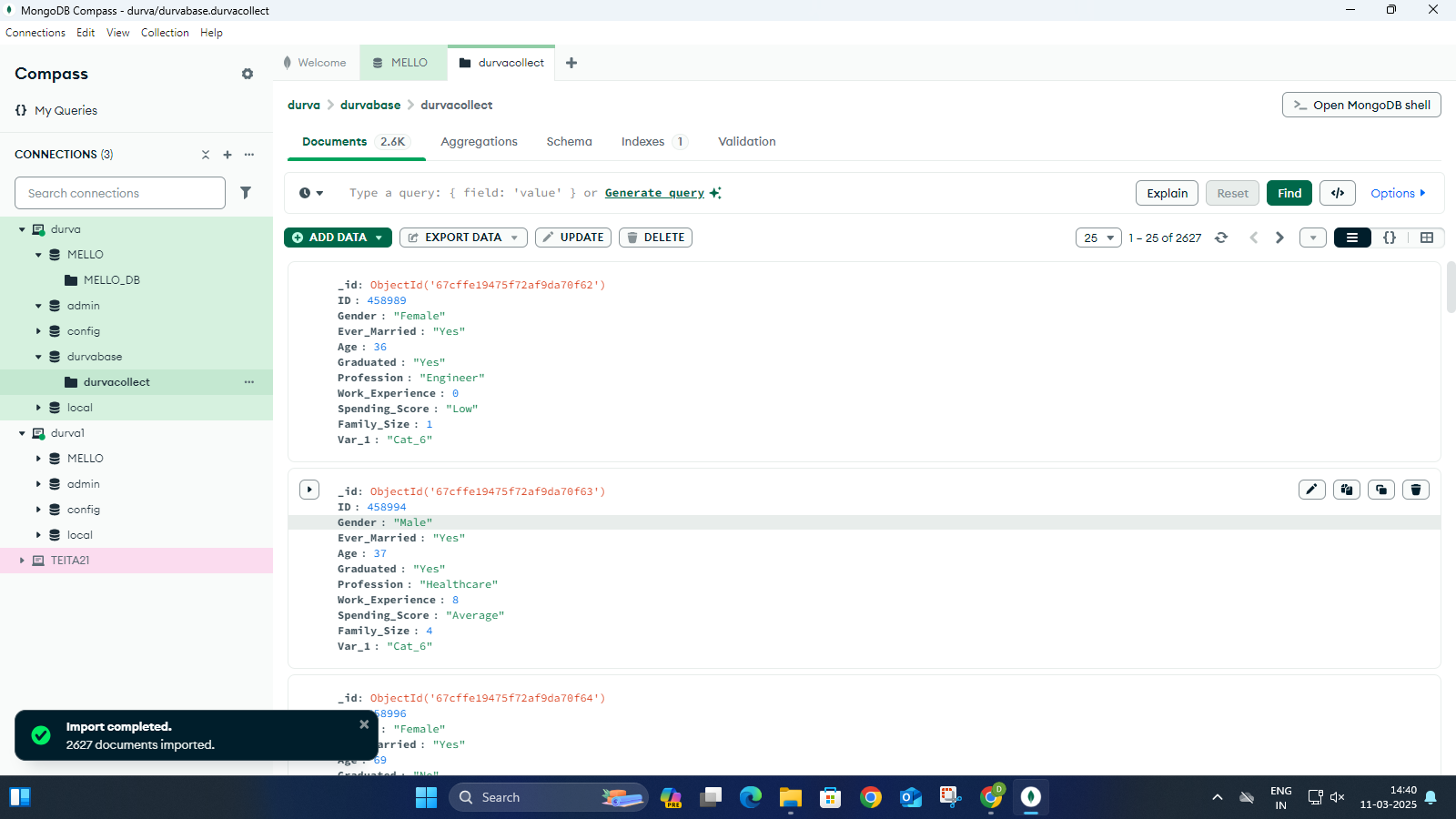
1. <https://docs.mongodb.com/manual/>
2. MongoDB in Action, Second Edition[,](https://freepdf-books.com/beginning-javascript-with-dom-scripting-and-ajax-2nd-edition-pdf-free-download/) by Kyle Banker, Peter Bakkum
3. <https://hevodata.com/learn/mongodb-compass/>
4. https://www.digitalocean.com/community/tutorials/how-to-use-mongodb-compass

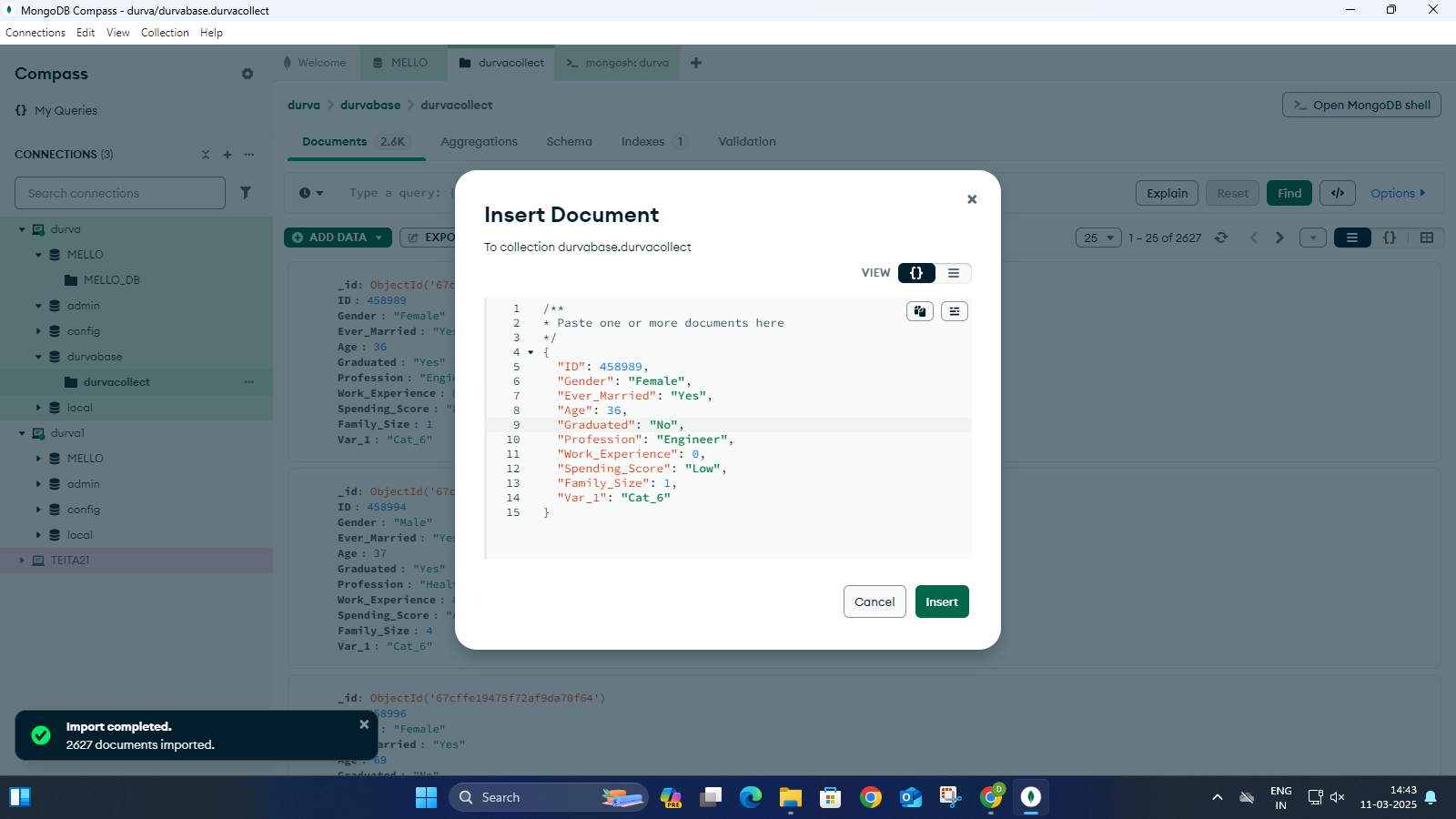
Creating a new Connection:



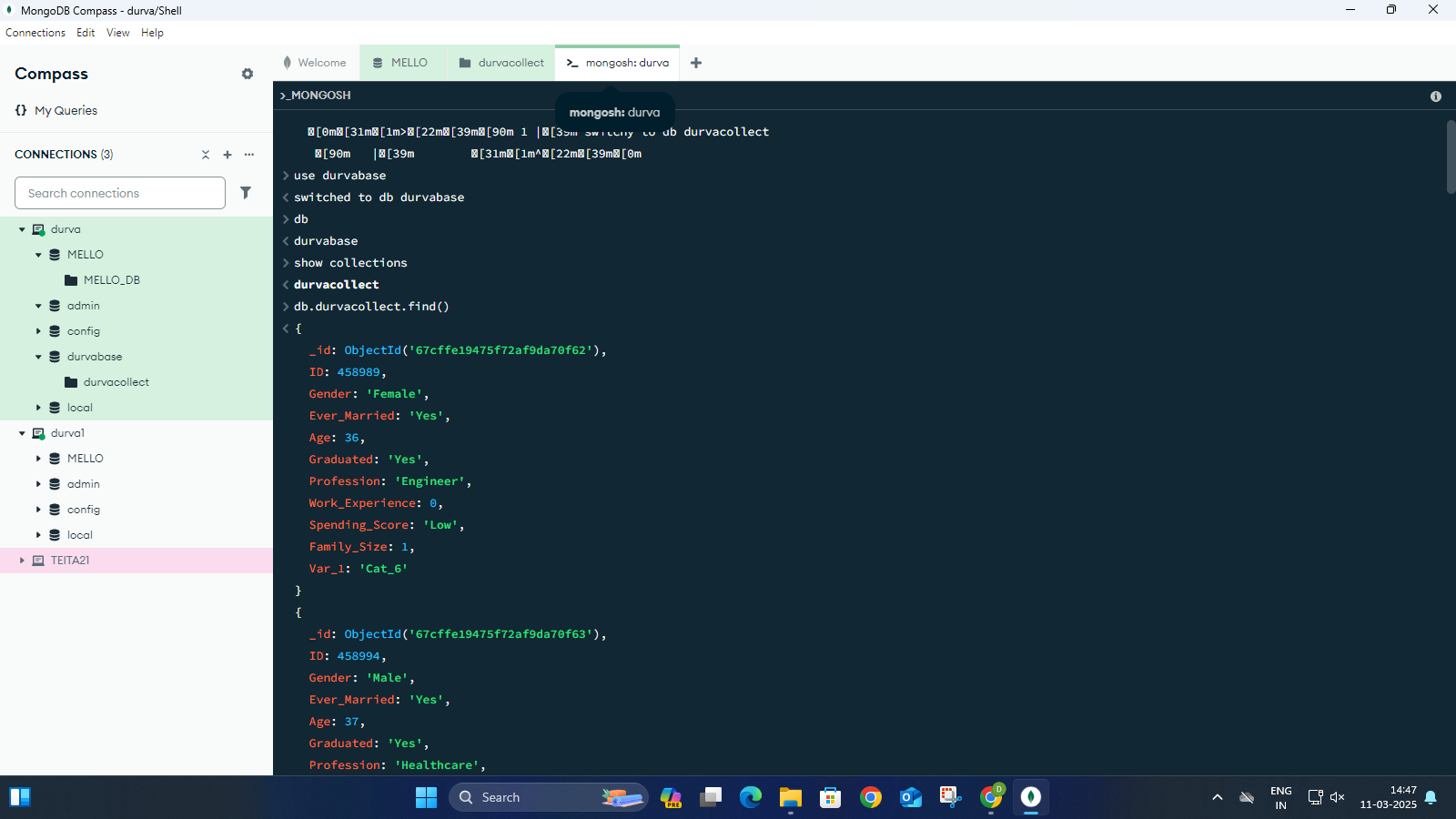
Creating a new database:

IMPORTING A CSV FILE:

Viewing the imported dataset:

MODIFYING THE DATA:

EDITING THROUGH THE SHELL:



USING THE CMD: 