**Practical No : 1**

**AIM :** Practical of Data Collection, Data Curation and Management for large scale data system (such as MongoDB).

**Theory :**

MongoDB is an open source NoSQL database management program. NoSQL is used as an alternative to traditional relational databases. NoSQL databases are quite useful for working with large sets of distributed data. MongoDB is a tool that can manage document-oriented information, store or retrieve information.

A collection is a grouping of MongoDB documents. Documents within a collection can have different fields. A collection is the equivalent of a table in a relational database system. A collection exists within a single database.

**MongoDB Features**

* Ad-hoc queries for optimized, real-time analytics. ...
* Indexing appropriately for better query executions. ...
* Replication for better data availability and stability. ...
* Sharding. ...
* Load balancing.

**Create Operations**

Create or insert operations add new [documents](https://www.mongodb.com/docs/manual/core/document/#std-label-bson-document-format) to a [collection](https://www.mongodb.com/docs/manual/core/databases-and-collections/#std-label-collections). If the collection does not currently exist, insert operations will create the collection.

MongoDB provides the following methods to insert documents into a collection:

* [db.collection.insertOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.insertOne/#mongodb-method-db.collection.insertOne)
* [db.collection.insertMany()](https://www.mongodb.com/docs/manual/reference/method/db.collection.insertMany/#mongodb-method-db.collection.insertMany)

In MongoDB, insert operations target a single [collection](https://www.mongodb.com/docs/manual/reference/glossary/#std-term-collection). All write operations in MongoDB are [atomic](https://www.mongodb.com/docs/manual/core/write-operations-atomicity/) on the level of a single [document.](https://www.mongodb.com/docs/manual/core/document/)

**Read Operations**

Read operations retrieve [documents](https://www.mongodb.com/docs/manual/core/document/#std-label-bson-document-format) from a [collection](https://www.mongodb.com/docs/manual/core/databases-and-collections/#std-label-collections); i.e. query a collection for documents. MongoDB provides the following methods to read documents from a collection:

* [db.collection.find()](https://www.mongodb.com/docs/manual/reference/method/db.collection.find/#mongodb-method-db.collection.find)

You can specify [query filters or criteria](https://www.mongodb.com/docs/manual/tutorial/query-documents/#std-label-read-operations-query-argument) that identify the documents to return.

**Update Operations**

Update operations modify existing [documents](https://www.mongodb.com/docs/manual/core/document/#std-label-bson-document-format) in a [collection](https://www.mongodb.com/docs/manual/core/databases-and-collections/#std-label-collections). MongoDB provides the following methods to update documents of a collection:

* [db.collection.updateOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.updateOne/#mongodb-method-db.collection.updateOne)
* [db.collection.updateMany()](https://www.mongodb.com/docs/manual/reference/method/db.collection.updateMany/#mongodb-method-db.collection.updateMany)
* [db.collection.replaceOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.replaceOne/#mongodb-method-db.collection.replaceOne)

In MongoDB, update operations target a single collection. All write operations in MongoDB are [atomic](https://www.mongodb.com/docs/manual/core/write-operations-atomicity/) on the level of a single document.

You can specify criteria, or filters, that identify the documents to update. These [filters](https://www.mongodb.com/docs/manual/core/document/#std-label-document-query-filter) use the same syntax as read operations.

**Delete Operations**

Delete operations remove documents from a collection. MongoDB provides the following methods to delete documents of a collection:

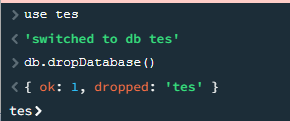
* [db.collection.deleteOne()](https://www.mongodb.com/docs/manual/reference/method/db.collection.deleteOne/#mongodb-method-db.collection.deleteOne)
* [db.collection.deleteMany()](https://www.mongodb.com/docs/manual/reference/method/db.collection.deleteMany/#mongodb-method-db.collection.deleteMany)

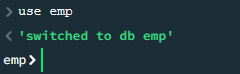
In MongoDB, delete operations target a single [collection](https://www.mongodb.com/docs/manual/reference/glossary/#std-term-collection). All write operations in MongoDB are [atomic](https://www.mongodb.com/docs/manual/core/write-operations-atomicity/) on the level of a single document.

You can specify criteria, or filters, that identify the documents to remove. These [filters](https://www.mongodb.com/docs/manual/core/document/#std-label-document-query-filter) use the same syntax as read operations.

**CODE:**

To Create and delete DB tes

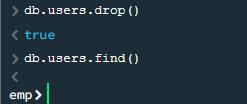




Create Collection using db.createCollection(“collectionname”)

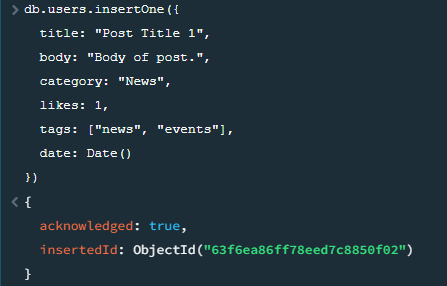


Delete collection



Insert data into collections

Using one input



 Using many inputs

db.posts.insertMany([

  {

    title: "Post Title 2",

    body: "Body of post.",

    category: "Event",

    likes: 2,

    tags: ["news", "events"],

    date: Date()

  },

  {

    title: "Post Title 3",

    body: "Body of post.",

    category: "Technology",

    likes: 3,

    tags: ["news", "events"],

    date: Date()

  },

  {

    title: "Post Title 4",

    body: "Body of post.",

    category: "Event",

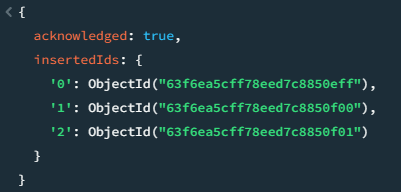
    likes: 4,

    tags: ["news", "events"],

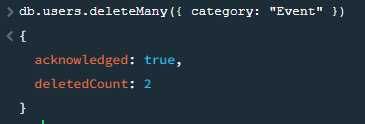
    date: Date()

  }

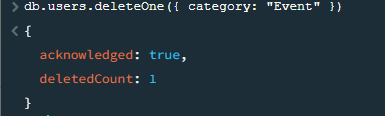
])



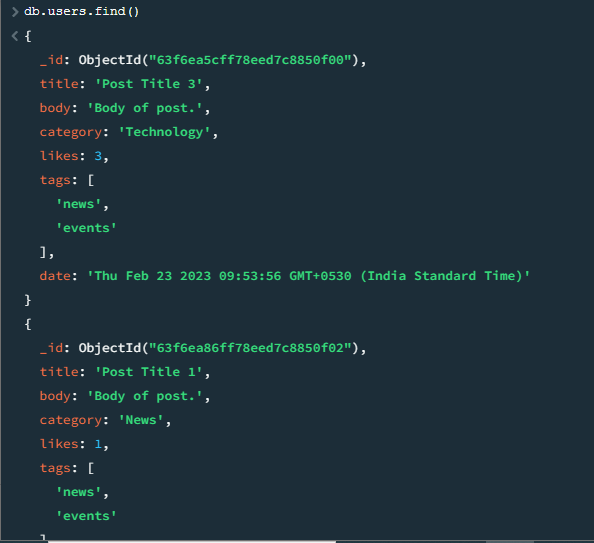
Delete multiple data using category



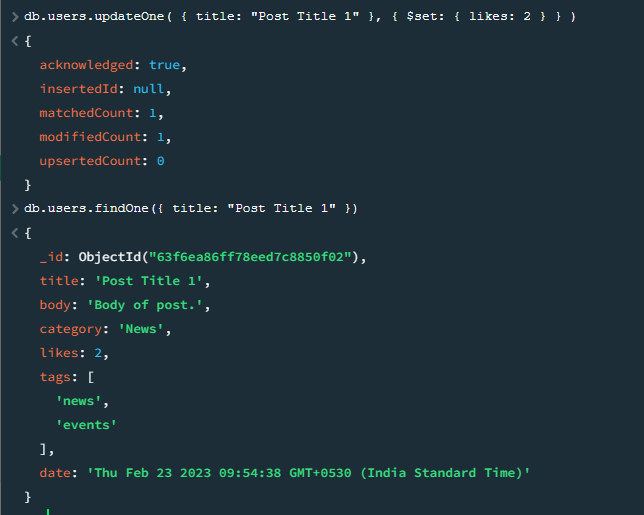
Delete single data



To view all the data use th find()



Update the likes



**Conclusion :** MongoDB is a non-relational document database that provides support for JSON-like storage. The MongoDB database has a flexible data model that enables you to store unstructured data, and it provides full indexing support, and replication with rich and intuitive APIs.

e